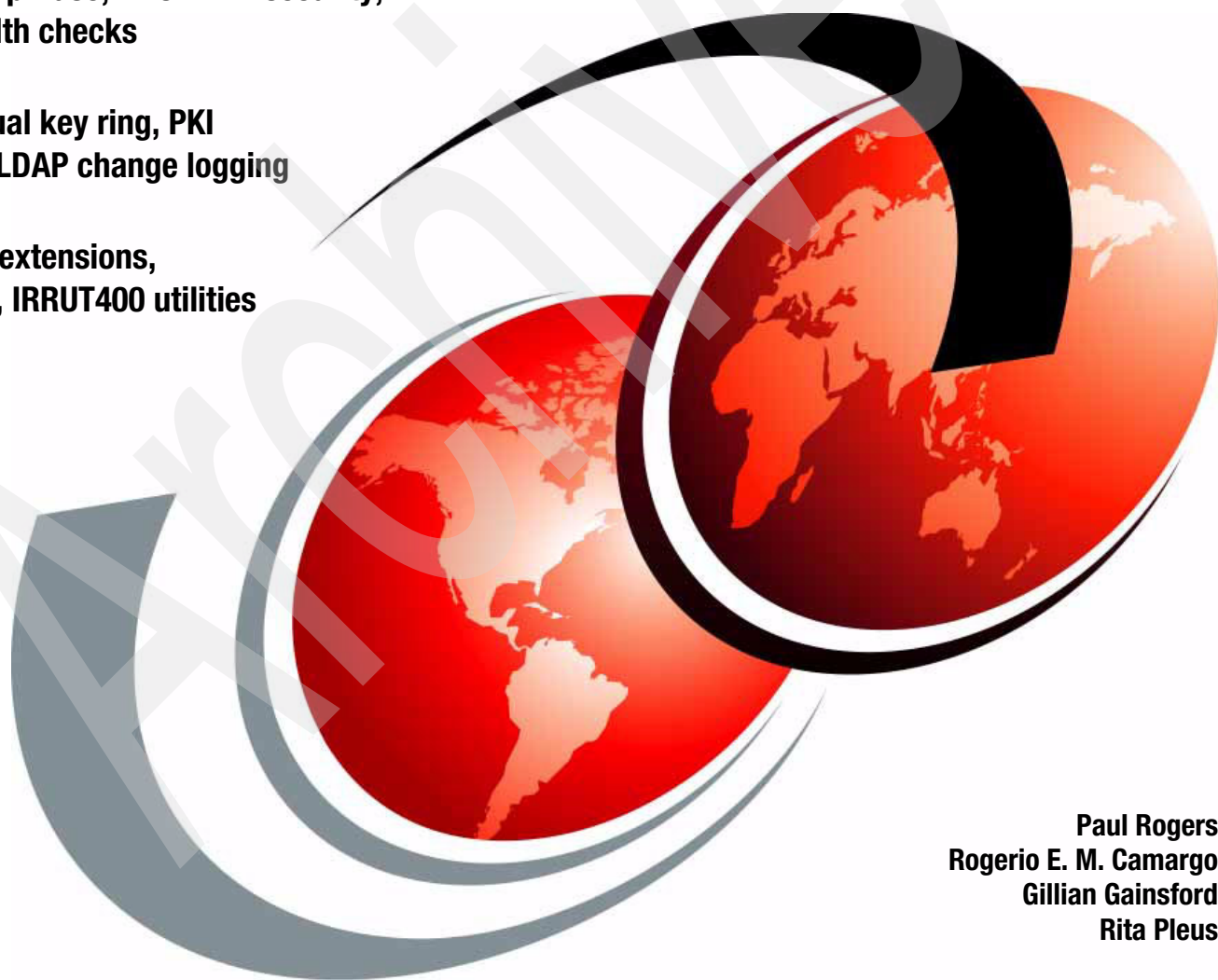


# z/OS Version 1 Release 8 RACF Implementation

Password phrase, RACF DB2 security,  
RACF health checks

RACF virtual key ring, PKI  
Services, LDAP change logging

Template extensions,  
IRRUT200, IRRUT400 utilities



Paul Rogers  
Rogerio E. M. Camargo  
Gillian Gainsford  
Rita Pleus

# Redbooks





International Technical Support Organization

**z/OS Version 1 Release 8 RACF Implementation**

February 2007

Archived

**Note:** Before using this information and the product it supports, read the information in “Notices” on page vii.

Archived

**First Edition (February 2007)**

This edition applies to Version 1 Release 8 of z/OS (5694-A01), Version 1 Release 8 of z/OS.e (5655-G52), and to all subsequent releases and modifications until otherwise indicated in new editions.

**© Copyright International Business Machines Corporation 2007. All rights reserved.**

Note to U.S. Government Users Restricted Rights -- Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

# Contents

<b>Notices</b> .....	vii
Trademarks .....	viii
<b>Preface</b> .....	ix
The team that wrote this redbook. ....	ix
Become a published author .....	x
Comments welcome. ....	x
<b>Chapter 1. RACF Version 1 Release 8</b> .....	1
1.1 Overview of RACF enhancements .....	2
1.2 Password phrase support .....	2
1.3 New RACF health checks .....	3
1.4 Enhancements with IRRUT200 and IRRUT400 .....	3
1.5 LDAP change log .....	3
1.6 Digital certificate support enhancements .....	4
1.7 SAF identity token .....	4
1.8 z/OS DB2 Version 8 support .....	4
1.9 Remote authorization and auditing .....	5
1.10 IRRSDA00 enhancements .....	5
<b>Chapter 2. Password phrase</b> .....	7
2.1 Password phrase benefits .....	8
2.1.1 Password phrase concepts .....	8
2.2 Password phrase and password .....	8
2.3 How the password phrase works .....	9
2.3.1 Password phrase rules .....	9
2.3.2 New password phrase ICHPWX11 exit .....	9
2.3.3 Password phrase change interval .....	10
2.4 RACF commands and password phrase .....	11
2.5 RACF remote sharing facility (RRSF) .....	14
2.5.1 Password phrase synchronization via PWSYNC .....	14
2.6 Password phrase and SETROPTS PASSWORD options .....	15
2.7 Password phrase auditing .....	17
2.8 Protected user IDs and password phrase .....	17
2.9 Providing the ability to reset password phrases .....	17
2.10 RACF utilities changes .....	18
2.10.1 RACF SMF data unload utility (IRRADU00) .....	18
2.10.2 RACF data security monitor (DSMON) .....	19
2.10.3 RACF database unload utility program (IRRDBU00) .....	19
2.11 New and changed RACF messages .....	19
<b>Chapter 3. Availability improvements for IRRUT200 and IRRUT400</b> .....	21
3.1 Synchronized copy with IRRUT200 .....	22
3.1.1 Pre-z/OS V1R8 implementation .....	22
3.1.2 Synchronized copy solution .....	22
3.1.3 Interaction, dependencies, and migration considerations .....	25
3.2 Safety features for IRRUT200 and IRRUT400 .....	25
3.2.1 Safety features with z/OS V1R8 .....	26
3.2.2 Safety feature implementation examples .....	26

3.2.3 Interaction, dependencies, and migrations considerations . . . . .	28
3.3 Publication updates . . . . .	28
3.3.1 Publications . . . . .	29
3.3.2 Changed IRRUT200 messages . . . . .	29
3.3.3 New IRRUT200 messages . . . . .	29
3.3.4 New and changed IRRUT400 messages . . . . .	29
<b>Chapter 4. RACF and the DB2 access control module . . . . .</b>	<b>31</b>
4.1 Previous DB2 versions . . . . .	32
4.1.1 Security implementation . . . . .	32
4.1.2 Expanding RACF protection to DB2 objects . . . . .	33
4.2 The RACF access control module - DSNXRXAC . . . . .	33
4.2.1 Modifying the RACF access control module . . . . .	35
4.2.2 Activating the RACF access control module . . . . .	36
4.2.3 Restarting DB2 with the RACF access control module . . . . .	39
4.3 Protecting DB2 objects with RACF profiles . . . . .	41
4.3.1 DB2 object privileges . . . . .	42
4.3.2 Mapping DB2 authorization checks . . . . .	43
4.3.3 Preventing cascading revoke . . . . .	45
4.4 Authorization checking . . . . .	48
4.5 Auditing considerations . . . . .	51
4.5.1 Debugging considerations . . . . .	52
4.6 Multilevel security . . . . .	52
4.6.1 DB2 and multilevel security . . . . .	55
4.6.2 Multilevel security with row-level granularity . . . . .	56
<b>Chapter 5. RACF virtual key ring support . . . . .</b>	<b>57</b>
5.1 RACF and key rings . . . . .	58
5.1.1 Secure Sockets Layer . . . . .	58
5.1.2 R_datalib (IRRSDL00) callable services . . . . .	59
5.1.3 What is a virtual key ring . . . . .	59
5.1.4 Problems before z/OS V1R8 . . . . .	59
5.1.5 RACF virtual key ring benefits . . . . .	59
5.1.6 How to use a virtual key ring . . . . .	59
5.1.7 Virtual key ring usage and invocation . . . . .	60
5.1.8 Virtual key ring implementation . . . . .	60
5.1.9 Real key ring and virtual key ring . . . . .	61
5.2 Related publications . . . . .	61
<b>Chapter 6. PKI Services . . . . .</b>	<b>63</b>
6.1 Introduction to PKI . . . . .	64
6.2 Overview of PKI Services . . . . .	64
6.2.1 Basic components of PKI Services and related products . . . . .	64
6.3 PKI Services multiple CAs overview . . . . .	65
6.3.1 z/OS V1R8 enhancements . . . . .	66
6.3.2 Loosely coupled CA examples . . . . .	66
6.3.3 Setup for PKI Services . . . . .	67
6.4 PKI Services support for SCEP . . . . .	68
6.4.1 PKI Services SCEP overview . . . . .	69
6.4.2 PKI Services usage considerations . . . . .	70
6.4.3 Using PKI Services utilities . . . . .	71
6.4.4 Preregistration rules . . . . .	71

<b>Chapter 7. RACF health checks</b> .....	75
7.1 IBM Health Checker for z/OS .....	76
7.1.1 Health checker overview .....	76
7.1.2 Flow of IBM Health Checker for z/OS .....	78
7.1.3 Security of IBM Health Checker for z/OS .....	79
7.1.4 User interface to manage checks .....	80
7.1.5 Using SDSF panels .....	80
7.1.6 Using (E)JES panels .....	85
7.1.7 Health Checker for z/OS commands via MODIFY command .....	86
7.1.8 HZSPRMxx parmlib member and policies .....	89
7.1.9 Policy statements .....	89
7.1.10 Categories to manage and display information .....	92
7.1.11 Criteria for the checks .....	93
7.2 Common features of all RACF checks .....	94
7.3 New RACF checks .....	94
7.3.1 Check RACF_<class-name>_ACTIVE .....	94
7.3.2 Check RACF_IBMUSER_REVOKED .....	97
7.4 Enhanced RACF checks .....	98
7.4.1 Check RACF_GRS_RNL .....	98
7.4.2 Check RACF_SENSITIVE_RESOURCES .....	102
<b>Chapter 8. LDAP change logging</b> .....	109
8.1 LDAP overview .....	110
8.2 Change log processing prior to z/OS V1R8 .....	112
8.3 Change log processing enhancements in z/OS V1R8 .....	112
8.4 Activating LDAP change notification .....	113
8.5 Password enveloping enhancements .....	114
8.6 Change logging of password changes .....	116
<b>Chapter 9. Template and profile extensions</b> .....	119
9.1 RACF database template extensions .....	120
9.1.1 Applications that read the RACF database directly .....	120
9.1.2 Migration considerations .....	122
9.2 Enhancements made to RACF profiles .....	123
9.2.1 Allowing or disallowing generics in static classes .....	123
9.2.2 Allowing or disallowing generics in dynamic classes .....	123
9.2.3 New messages with z/OS V1R8 .....	126
9.2.4 Migration and coexistence considerations .....	127
9.3 IRRDPI00 LIST command granularity .....	127
9.4 KERBLINK class enhancement .....	128
9.4.1 Migration considerations for KERBLINK class .....	129
9.5 OMVS FILEPROC MAX change .....	129
<b>Related publications</b> .....	131
IBM Redbooks .....	131
Other publications .....	131
Online resources .....	131
How to get IBM Redbooks .....	132
Help from IBM .....	132
<b>Index</b> .....	133

Archived



# Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

*IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785 U.S.A.*

**The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:** INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.


This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

## COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

## Trademarks

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both:

Redbooks (logo) ™

z/OS®

z/VM®

zSeries®

AFP™

DB2 Universal Database™

DB2®

Infoprint®

IBM®

MVS™

OS/390®

Redbooks™

RACF®

REXX™

S/390®

Tivoli®

WebSphere®

The following terms are trademarks of other companies:

Java, and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, or service names may be trademarks or service marks of others.

# Preface

This IBM® Redbook describes the implementation of RACF® in z/OS® Version 1 Release 8. This release continues to deliver industry leadership for security. Improvements have been introduced to further enhance the security-rich environment z/OS users rely on. These enhancements include:

- ▶ RACF support for virtual key rings to treat the collection of all the certificates owned by one user ID, including the SITE and CERTAUTH reserved user IDs, as an independent key ring. The use of the CERTAUTH virtual key ring will help to eliminate the need to manually create multiple real key rings for SSL-enabled z/OS client applications such as FTP.
- ▶ RACF template extensions allow templates to expand beyond their current 4K size.
- ▶ RACF supports the use of passwords longer than eight characters, now called password phrases.
- ▶ The RACF access control module exit, DSNXRXAC, has changed substantially with DB2® version 9. A RACF administrator can now define a security rule before an object is created and preserve the rule for a dropped object. In addition, RACF general resources for member and group profiles can be used by an installation to protect multiple DB2 resources with a single RACF profile.
- ▶ A new parameter on the IRRUT200 utility tells the utility to activate the backup data set printed to as output. This is accomplished by the utility internally issuing an RVAR Y ACTIVE for the backup data set after the copy is complete. IRRUT200 and IRRUT400 utilities now check whether their output data sets are active primary or backup RACF data sets on this system.
- ▶ New RACF health checks are introduced.
- ▶ RACF in z/OS V1R8 provides a solution to some functional gaps in the way that change logging of RACF profile updates were reflected in z/OS LDAP, and an enhancement is made to LISTUSER to demonstrate whether password enveloping is enabled for a user.

In addition to describing the new features, this book includes detailed steps for implementing these enhancements. It explains how to configure them for your installation and how to use them to increase the security of your environment.

## The team that wrote this redbook

This redbook was produced by a team of specialists from around the world working at the International Technical Support Organization, Poughkeepsie Center.

**Paul Rogers** is a Consulting IT Specialist at the International Technical Support Organization, Poughkeepsie Center. He writes extensively and teaches IBM classes worldwide on various aspects of z/OS, z/OS UNIX®, JES3, and Infoprint® Server. Before joining the ITSO 19 years ago, Paul worked in the IBM Installation Support Center (ISC) in Greenford, England for 7 years providing OS/390® and JES support for IBM EMEA and also in the Washington Systems Center for 3 years. He has worked for IBM for 39 years.

**Rogério Eugenio Malaquias Camargo** is a Senior IT Security Specialist in IBM Global Services in IBM Brasil. He holds a degree in Computer Science and has 10 years of

experience in IT Security Mainframe platforms. His areas of expertise include User Management, Security Process, Security Compliance, and Technical Security Tests.

**Gillian Gainsford** is a Security Systems Programmer in IBM Global Services in New Zealand. She has 23 of experience in the mainframe systems programming field and has worked at IBM for eight years. Gillian holds a Bachelor of Arts degree from the University of Auckland. Her area of expertise is the compliance and security of z/OS systems using both RACF and Ca-ACF2 as well as the inherent security features of z/OS and its subsystems. Prior to joining IBM Gillian worked as Senior Systems programmer in the banking, manufacturing, and airline industries.

**Rita Pleus** is a Senior IT Specialist in IBM Global Technology Services in IBM Germany. She has 20 years of IT experience in a variety of areas, including systems programming and operations management. Before joining IBM in 2001, she worked for a German S/390® customer. Rita holds a degree in Computer Science from the University of Applied Sciences in Dortmund. Her areas of expertise include z/OS, its subsystems, and systems management. She was one of the authors of ABCs of z/OS System Programming Volume 3 and Volume 6, and teaches the z/OS Security Server fundamentals class in Germany.

## Become a published author

Join us for a two- to six-week residency program! Help write an IBM Redbook dealing with specific products or solutions, while getting hands-on experience with leading-edge technologies. You'll team with IBM technical professionals, Business Partners and/or customers.

Your efforts will help increase product acceptance and customer satisfaction. As a bonus, you'll develop a network of contacts in IBM development labs, and increase your productivity and marketability.

Find out more about the residency program, browse the residency index, and apply online at:

[ibm.com/redbooks/residencies.html](http://ibm.com/redbooks/residencies.html)

## Comments welcome

Your comments are important to us!

We want our Redbooks™ to be as helpful as possible. Send us your comments about this or other Redbooks in one of the following ways:

- Use the online **Contact us** review redbook form found at:

[ibm.com/redbooks](http://ibm.com/redbooks)

- Send your comments in an e-mail to:

[redbook@us.ibm.com](mailto:redbook@us.ibm.com)

- Mail your comments to:

IBM Corporation, International Technical Support Organization  
Dept. HYTD Mail Station P099  
2455 South Road  
Poughkeepsie, NY 12601-5400

# RACF Version 1 Release 8

This chapter contains an overview of the enhancements in RACF V1R8. It describes the following functional changes made to RACF in this release:

- ▶ Password phrase support
- ▶ New RACF health checks
- ▶ Enhancements with IRRUT200 and IRRUT400
- ▶ LDAP change log
- ▶ PKI enhancements
- ▶ SAF identity token
- ▶ z/OS DB2 version 9 support
- ▶ Remote authorization and auditing
- ▶ IRRSDA00 enhancements

## 1.1 Overview of RACF enhancements

In z/OS Version 1 Release 8, z/OS continues to deliver industry leadership for security. Improvements that deliver the kind of security-rich environment that has made z/OS an industry leader include:

- ▶ Support for defining Intrusion Detection Services (IDS) policies in a policy agent configuration file as well as an LDAP server. This provides an IDS policy solution that is consistent with other policy types for those installations that do not have an LDAP infrastructure in place or that prefer using configuration files instead of LDAP.
- ▶ RACF support for password phrases from 14 to 100 characters in length, in addition to the current support for passwords. Password phrases allow for an exponentially greater number of possible combinations of characters and numbers than do passwords.
- ▶ New options for securing tape data sets using the system authorization facility (SAF). These options allow you to:
  - Define profiles to protect data sets on tape using the DATASET class without the need to activate the TAPEDSN option or the TAPEVOL class
  - Specify that all data sets on a tape volume should have common authorization
  - Specify whether users are authorized to overwrite existing files on a tape volume
- ▶ Support for the advanced encryption standard (AES) algorithm for IP Security with a 128-bit key length.
- ▶ Support for SAF identity tokens. The support for SAF identity tokens provides exploiters with increased user accountability and auditability of resources by providing end-to-end auditing that tracks identities used for initial authentication and those used on the current platform.
- ▶ Digital certificate support enhancements, including:
  - RACF support for virtual key rings. This support treats the collection of all the certificates owned by one user ID, including the SITE and CERTAUTH reserved user IDs, as an independent key ring. The use of the CERTAUTH virtual key ring is intended to help eliminate the need to manually create multiple real key rings for SSL-enabled z/OS client applications such as FTP.
  - Public Key Infrastructure (PKI) services support for multiple certificate authorities (CA) on a single image.
  - Adding SCEP support to PKI services. A simple certificate enrollment protocol (SCEP) allows SCEP-enabled clients (for example, a router) to request certificates.

## 1.2 Password phrase support

In z/OS V1R8, profile extension focuses on improving or making more flexible and relevant the security rules as implemented by RACF today. Password phrase support and custom fields support both lead toward this goal. Password phrase support provides an alternative to traditional passwords. This alternative, a password phrase, allows a much longer value than a password, as well as a larger character set. A password phrase is a character string, made up of mixed case letters, numbers, and special characters, including blanks. The intention of the password phrase is to have something that is long enough to provide security, but easy enough to remember that it will not be written down. It is unlikely that a random string of characters longer than eight characters can be memorized easily; therefore, a password phrase is more likely to be made up of words. Adding numbers and special characters makes the phrase more secure. RACF now supports password phrases from 14 to 100 characters in

length, and enforces a basic set of rules to increase the strength of the password phrase. Since a user ID can have both a password and a password phrase, the same user ID can be used for both traditional applications that accept a password, and for new applications that take advantage of the password phrase infrastructure.

## 1.3 New RACF health checks

RACF provides the following new health checks:

- ▶ Examine additional key security controls on the system, such as the current link list libraries, the current parmlib data sets, and key general resources.
- ▶ Examine the IBMUSER user ID and ensure that it is revoked.
- ▶ Examine key classes (such as TAPEVOL) and ensure that the classes are active.
- ▶ Support the verbose flag, which when set to ON causes the RACF\_FRS\_RNL check to list all of the ENQ names (just like the DEBUG option does currently).
- ▶ Allow installations to define their own resources to a RACF-supplied check.

## 1.4 Enhancements with IRRUT200 and IRRUT400

RACF has made availability improvements in z/OS V1R8, which can help to identify and correct problems that have resulted in past system outages. The problems addressed in this release are:

- ▶ Clients needing to create a consistent backup copy of the primary RACF data set and activate the backup without loss of synchronization between the two data sets. In the past, IRRUT200 did not allow this because the primary RACF database could be updated between the copy process and the activation of the backup database, thus resulting in an error. IRRUT400 with the LOCKINPUT option could create a consistent RACF backup database. However, it could cause applications attempting to access the locked database to fail.
- ▶ Simple errors in the DD statements for IRRUT200 or IRRUT400 could result in an active RACF data set being an output data set for these utilities. The utilities should detect this when possible and prevent the overwriting of an active RACF data set.

### RAS enhancements

To solve these problems, RACF has made the following availability improvements:

- ▶ A new parameter on the IRRUT200 utility will tell the utility to activate the backup data set pointed to as output. This will be accomplished by the utility internally issuing an RVARY ACTIVE for the backup data set after the copy is complete.
- ▶ IRRUT200 and IRRUT400 utilities will check whether their output data sets are active primary or backup RACF data sets on this system. If they are, the utility will fail with an error message.

## 1.5 LDAP change log

RACF currently supports event notification if a USER profile changes using the z/OS LDAP change log. Group change logging extends this support to GROUP profiles, including group connection information. This closes a functional gap such that all RACF information that can

be managed using the LDAP SDBM back-end will not participate in the LDAP change logging function. In addition, two minor functional gaps were also closed with regard to the related password enveloping function, as follows:

- ▶ LDAP change log entries are now created for any password change, not just those which were enveloped.
- ▶ The **LISTUSER** command has been enhanced to report on the presence of a password envelope for a user. Previously, an administrator had to run the RACF database unload utility (IRRDBU00) to obtain this information.

## 1.6 Digital certificate support enhancements

In z/OS V1R8, digital certificate support is being enhanced with the following capabilities:

- ▶ Virtual key rings: In the past, SSL-enabled applications required creating a key ring for each unique user ID along with any CA certificates. This situation propagated, when in most cases the same set of CA certificates would be used for each user ID. Thus these key rings would all be replicas.

To solve this problem, RACF now treats all the certificates installed under a given user ID as a virtual key ring. This key ring is created when the user ID is added, and destroyed when the user ID is deleted.

- ▶ Enabling multiple CA support for PKI Services: This function now lifts the restriction that prevented more than one instance of the PKI Services daemon from being started simultaneously on a single MVS™ image. This allows PKI services customers to establish multiple certificate authorities on a single MVS image.
- ▶ Add SCEP support to PKI services: Simple certificate enrollment protocol (SCEP) allows SCEP-enabled clients to request certificates by sending messages to a certificate authority using the HTTP protocol. Adding SCEP support allows PKI services to accept, decrypt, and respond to the various SCEP messages, and supports both the manual and automatic enrollment modes as defined in the standard. Manual enrollment requires the CA administrator to manually approve requests. With automatic enrollment, certificate requests are auto-approved and fulfilled synchronously, based on the requestor's knowledge of a predetermined secret, the challenge password.

## 1.7 SAF identity token

SAF identity token now provides increased user accountability and audit resources by providing end-to-end auditing that tracks the identity initially used for authentication as well as the identity on the current platform. This support is especially valuable to customers maintaining heterogeneous environments, where requests and entry points to network resources come from a variety of platforms.

## 1.8 z/OS DB2 Version 8 support

RACF supplies a security exit that enables SAF/RACF services to be utilized for DB2 security, effectively allowing DB2 security to be centralized within the scope of the system security administrator. RACF now extends the existing support to provide conditional access support, which enables the use of the DB2 role to authorize a user to a RACF protected DB2 object. Further, in support of DB2's trusted context, the identity, which is conveyed to z/OS DB2, may not be a z/OS RACF defined user. RACF provides an identity mapping plug which



enables z/OS DB2 foreign users to a z/OS RACF defined identity, which is needed for the resolution of access control decisions. This function provides RACF security support to the DB2 V8 community by using the new trusted channel role.

## 1.9 Remote authorization and auditing

RACF now provides the capability of processing remote authorization and auditing requests as the z/OS security server. These features will employ a standard LDAP protocol to enable requests from various system platforms, ensure an auditable level of trust between the z/OS security server and requesting applications, use familiar SMF logging capability for recording audit data, and support the unloading of that data. Customers increasingly look for value in a centralized directory such as z/OS LDAP to locate and manage people, objects, and services for the enterprise. Middleware application hosting environments, WebSphere® and Tivoli®, have embraced LDAP for that purpose. Leveraging these new z/OS LDAP extended operations, middleware and customer applications can achieve distributed authorization and auditing function that can be managed by the z/OS security administrator.

## 1.10 IRRSDA00 enhancements

RACF provides an enhancement to IRRSDA00 to support using RACROUTE REQUEST=FASTAUTH when the RACF profile is RACLISTed. Currently, CIM Server is using `__check_resource_auth_np` to make sure that the CIM client can access CIM Server. The existing `__check_resource_auth_np` logic translates to UNIX System Services invoking IRRSDA00 using RACROUTE REQUEST=AUTH function.

The RACROUTE REQUEST=AUTH function is a long pathlength function. Since the CIM Server RACF profile is RACLISTed, changing IRRSDA00 to recognize the RACLISTed profile to do a FASTAUTH check will have performance benefits for all `__check_resource_auth_np()` invokers checking on RACLISTed profiles.

The IRRSDA00 has added logic to create an ACEE for the input user ID for the FASTAUTH check on the RACLISTed profiles. This change gives better performance on `__check_resource_auth_np` when the RACF profile is RACLISTed.

Archived

## Password phrase

This chapter discusses the new RACF feature *password phrase*, which is introduced with z/OS V1R8.

Significant improvements have been made to RACF password processing, and this chapter describes the following updates:

- ▶ Password phrase concepts
- ▶ Password phrase usage and invocation
- ▶ RACF commands for password phrase
- ▶ Updates to RACF utilities to support password phrase

## 2.1 Password phrase benefits

Users can have a password or a password phrase, or both, and therefore the same user ID can be used for both traditional applications that accept passwords, and for new applications that take advantage of the password phrase implementation.

### 2.1.1 Password phrase concepts

In z/OS V1R8, RACF has implemented the password phrase technology as an infrastructure to provide an alternative to traditional passwords, thereby improving system security authentication to better fit with distributed operation systems.

Your password might need to satisfy certain installation-defined rules. RACF is able to exploit the password phrase as a string of characters from 14 to 100 bytes in length, well beyond the previous standard password limit of 8 bytes on traditional passwords. Furthermore, password phrases can contain characters that are not allowed in a password, including blanks.

In addition to traditional passwords, you can also have an optional password phrase, which you can use instead of a password with applications that support password phrase technology.

Password phrases provide a security advantage over traditional passwords because they are long enough to withstand most hacking attempts.

**Note:** Although RACF allows setting a password phrase, currently no z/OS V1R8 components support the use of password phrase. Password phrase is an alternative to your password for verifying your identity; however, your installation might have applications that support password phrases.

## 2.2 Password phrase and password

Table 2-1 compares the characteristics and functions of password phrases and traditional passwords.

Table 2-1 Password and password phrase comparison

Functions	Traditional password	Password phrase
Password syntax rules	Able to establish up to 8 different password syntax rules	<ul style="list-style-type: none"><li>▶ Must be a text string of 14-100 characters</li><li>▶ Must not contain the user ID</li><li>▶ Must contain at least 2 alphabetic characters</li><li>▶ Must contain at least 2 non-alphabetic characters</li><li>▶ Must not contain more than 2 consecutive characters that are identical</li></ul>
Password EXIT	ICHPWX01	ICHPWX11
Where to use it	Traditional applications / systems in the Mainframe and Distributed operation system	Distributed operation system that accepts the password phrase technology

Functions	Traditional password	Password phrase
RACF commands involved	ADDUSER / ALTUSER and PASSWORD command	ADDUSER / ALTUSER and PASSWORD command
SETR PASSWORD MAXIMUM and MINIMUM INTERVAL	YES	YES
SETR PASSWORD HISTORY	YES	YES
SETR PASSWORD REVOKE	YES	YES
SETR PASSWORD MIXEDCASE	YES	NO
SETR PASSWORD WARNING	YES	NO
SETR PASSWORD RULE	YES	NO
AUDIT events	YES	YES
Can IRR.PASSWORD.RESET be used to reset it?	YES	YES
RACF Remote Sharing Facility (RRSF)	PWSYNC resource	PHRASESYNC resource
Automatic Password Direction (APD)	YES	YES

## 2.3 How the password phrase works

You can issue the PHRASE operand with the RACF ADDUSER or ALTUSER command to assign a password phrase for a user ID. The rules are set according to the RACF password phrase syntax rules and the ICHPWX11 exit. This enables the user to authenticate itself using a password phrase instead of a password when using an application that supports password phrase.

### 2.3.1 Password phrase rules

RACF enforces the following rules for a password phrase:

- ▶ Must be a text string of between 14 and 100 characters
- ▶ Must not contain the user ID in sequential upper case or sequential lower case characters
- ▶ Must contain at least 2 alphabetic characters (A-Z, a-z)
- ▶ Must contain at least 2 non-alphabetic characters (numerics, punctuation, special characters, or blanks)
- ▶ Must not contain more than 2 consecutive identical characters

These rules are hard-coded, so no SETROPTS options are necessary.

### 2.3.2 New password phrase ICHPWX11 exit

The installation has the option of using the new password phrase, ICHPWX11exit, to augment RACF functions when validating a new password phrase. The new password

phrase ICHPWX11 exit gains control when a new password phrase is processed through the RACF commands and functions:

- ▶ ADDUSER command
- ▶ ALTUSER command
- ▶ PASSWORD/PHRASE
- ▶ RACROUT=REQUEST=VERIFY

The ICHPWX11 exit enforces installation password rules in addition to the RACF rules; however, the new password phrase ICHPWX11 exit does not override RACF password phrase rules.

**Note:** If the new password phrase ICHPWX11 exit requires that the new password phrases contain only alphabetic characters, users will not be able to create new password phrases because the password phrase rules require at least two non-alphabetic characters.

RACF enforces a basic set of syntax rules to establish strength in password phrases. These syntax rules apply to all password phrases and you cannot alter or ignore them. However, you can add password phrase syntax rules to impose additional restrictions when your installation tailors the new password phrase ICHPWX11 exit.

If the ICHPWX11 exit exists, it can reject the password phrase. If the password phrase is accepted, it is made the user's current password phrase and, if the SETROPTS PASSWORD(HISTORY) command is in effect, the password phrase is added to the user ID's password phrase history.

### 2.3.3 Password phrase change interval

The password phrase must be changed after a certain interval of time to help ensure that it is known only by you. The interval is the same one that determines when you must change your password. The interval at which you must change your password and password phrase is specified in your SETROPTS PASSWORD INTERVAL suboperand, as shown in Figure 2-1.

```
PASSWORD PROCESSING OPTIONS:  
PASSWORD CHANGE INTERVAL IS 90 DAYS.
```

*Figure 2-1 Setropts password maximum change interval*

The password minimum change suboperand also applies to the password phrase. This is the minimum time that must pass between password phrase changes, as shown in Figure 2-2.

```
PASSWORD PROCESSING OPTIONS:  
PASSWORD MINIMUM CHANGE INTERVAL IS 5 DAYS.
```

*Figure 2-2 Setropts password minimum change interval*

When you change the SETROPTS PASSWORD(INTERVAL) value, the password interval set in each user's profile is not changed. If a user's interval value in the user's profile is different than the SETROPTS value, RACF expires the password phrase at the shorter interval of the two values, as shown in Figure 2-3 on page 11.

```

SETR LIST:
PASSWORD PROCESSING OPTIONS:
    PASSWORD CHANGE INTERVAL IS 90 DAYS.  <=====

LIST USER:
    USER=DEFAULT  NAME=USER  DEFAULT TEST  OWNER=SYS1
    DEFAULT-GROUP=SYS1  PASSDATE=00.000  PASS-INTERVAL=186

```

Figure 2-3 Difference between Setropts password interval and user ID's interval

## 2.4 RACF commands and password phrase

There are several RACF commands that now have modifications for password phrases. These commands are changed to have a password phrase suboperand.

### Add user ID with ADDUSER | AU

The ADDUSER command with the PHRASE suboperand specifies the user ID's initial password phrase, as shown in Figure 2-4. The password phrase is always set to expired, thus requiring the user to change it on initial use.

```

FUNCTION
ADDUSER (ADD USER PROFILE) using the password phrase suboperand:

    ADDUSER | AU
              (userid ...)
              PHRASE ('passphrase')

```

Figure 2-4 ADDUSER syntax command with the PHRASE suboperand

Every user that is assigned a password phrase must also have a password. In other words, a user cannot have only a password phrase. When you specify the PHRASE suboperand for a user without specifying a PASSWORD, the user is assigned the default password. When you specify the PHRASE suboperand with the NOPASSWORD option, an error message is issued indicating that the NOPASSWORD operand is ignored and the user is assigned the default password, as shown in Figure 2-5.

```

AU TEST1 NOPASSWORD PHRASE('CHECKING RACF COMMANDS')
NOPASSWORD OPERAND IGNORED

```

Figure 2-5 ADDUSER command with NOPASSWORD and PHRASE suboperand

If the PHRASE suboperand is omitted, no password phrase is assigned; however, if you enter the PHRASE suboperand without a password phrase value, you are prompted for a value, as shown in Figure 2-6. The value must be entered in quotes.

```

AU TEST1 PHRASE
ENTER password phrase -

```

Figure 2-6 ADDUSER command specifying a PHRASE without a value

**Note:** If the exit ICHPWX11 is present, it can reject the specified password phrase.

## List user profile with LISTUSER I LU

You can list the user profile to verify whether a password phrase is assigned to the user ID, and if one is, you can also verify the last password phrase change date through the PHRASEDATE field. The LISTUSER command output shows the following new fields:

- ▶ ATTRIBUTES=PASSPHRASE  
if a password phrase is assigned to the user ID, as shown in Figure 2-7.

```
USER=TEST1  NAME=UNKNOWN  OWNER=SYS1      CREATED=06.186
DEFAULT-GROUP=SYS1      PASSDATE=00.000 PASS-INTERVAL= 90 PHRASEDATE=06.186
ATTRIBUTES=PASSPHRASE
```

Figure 2-7 LISTUSER command output showing the PASSPHRASE attribute field

- ▶ PHRASEDATE field, where you can find the following parameter, as shown in Figure 2-8:
  - PHRASEDATE=nn.nnn is the last password phrase change date.
  - PHRASEDATE=00.000 if the user ID has an expired password phrase.
  - PHRASEDATE=N/A if the user ID has no password phrase.

```
USER=TEST1  NAME=UNKNOWN  OWNER=SYS1      CREATED=06.186
DEFAULT-GROUP=SYS1      PASSDATE=00.000 PASS-INTERVAL= 90 PHRASEDATE=06.186
ATTRIBUTES=PASSPHRASE

USER=TEST1  NAME=UNKNOWN  OWNER=SYS1      CREATED=06.186
DEFAULT-GROUP=SYS1      PASSDATE=00.000 PASS-INTERVAL= 90 PHRASEDATE=00.000
ATTRIBUTES=PASSPHRASE

USER=TEST1  NAME=UNKNOWN  OWNER=SYS1      CREATED=06.186
DEFAULT-GROUP=SYS1      PASSDATE=00.000 PASS-INTERVAL= 90 PHRASEDATE=N/A
ATTRIBUTES=NONE
```

Figure 2-8 LISTUSER command output showing the PHRASEDATE field

## Alter user ID with ALTUSER I ALU

The ALTUSER command with the PHRASE suboperand specifies the user ID's initial password phrase. The password phrase is always set to "expired" unless the NOEXPIRED operand is issued, thus requiring the user to change it on initial use.

The ALTUSER command with the PHRASE suboperand is used to perform the following actions:

- ▶ Add a password phrase to a user ID that does not have a password phrase
- ▶ Change a user ID's password phrase
- ▶ Remove a password phrase from a user ID that has one

Figure 2-9 on page 13 shows the RACF ALTUSER command syntax with the password phrase suboperand.



```

FUNCTION
ALTUSER (ALTER USER PROFILE) using the password phrase suboperand:

ALTUSER | ALU
         (userid ...)
         PHRASE('passphrase') | NOPHRASE

```

Figure 2-9 ALTUSER syntax command with the PHRASE and NOPHRASE suboperand

The NOPHRASE suboperand specifies that the user cannot use a password phrase for authentication. If a password phrase was previously set, it is cleared. The date of the last password phrase is also cleared from the user's profile.

The EXPIRED and NOEXPIRED suboperands also apply to password phrase.

If you specify PHRASE NOEXPIRED with the ALTUSER command, as shown in Figure 2-10, the user ID does not need to change the password phrase at the first logon.

```

ALU TEST1 PHRASE('PASSWORD $ password phrase') NOEXPIRED

```

Figure 2-10 ALTUSER syntax command with PHRASE and NOEXPIRED suboperand

However, it does not indicate the password phrase will never expire. If you want to set a password phrase that never expires, use the NOINTERVAL suboperand, as shown in Figure 2-11.

```

ALU TEST1 PHRASE('PASSWORD $ password phrase') NOEXPIRED

USER=TEST1 NAME=UNKNOWN OWNER=SYS1      CREATED=06.186
DEFAULT-GROUP=SYS1      PASSDATE=00.000 PASS-INTERVAL= 90 PHRASEDATE=06.186
ATTRIBUTES=PASSPHRASE

PASSWORD NOINTERVAL USER(TEST1)

USER=TEST1 NAME=UNKNOWN OWNER=SYS1      CREATED=06.186
DEFAULT-GROUP=SYS1      PASSDATE=00.000 PASS-INTERVAL=N/A PHRASEDATE=06.186
ATTRIBUTES=PASSPHRASE

```

Figure 2-11 Password phrase noexpired and nointerval

## PASSWORD | PHRASE | PW

This command is usually called the PASSWORD command even though the PHRASE suboperand has been added as an alias of the PASSWORD command.

The value specified by INTERVAL applies to the password as well as the password phrase. The USER keyword does not apply to password phrases. If the PHRASE keyword is specified with the USER keyword, the PHRASE keyword is ignored.

The new PASSWORD or PHRASE syntax command is shown in Figure 2-12 on page 14.

```

FUNCTION
PASSWORD OR PHRASE (SPECIFY USER PASSWORD OR PASSWORD PHRASE)

PASSWORD | PW | PHRASE
    AT( node .userid ...) | ONLYAT( node .userid ...)
    INTERVAL(change-interval) | NOINTERVAL
    PASSWORD(current-password new-password)
    PHRASE('current-passphrase' 'new-passphrase')
    USER(userid ...)

```

Figure 2-12 PASSWORD syntax command

## 2.5 RACF remote sharing facility (RRSF)

With RRSF there are two types of user ID association, peer and managed, as follows:

- ▶ **Peer association:** Allows either of the associated user IDs to direct commands to the other and allows the associated user IDs to synchronize their password phrases or passwords.
- ▶ **Managed association:** One of the user IDs is designated as the managing ID, and the other is designated as the managed ID. The managing user ID can direct commands to the managed ID, but the managed ID cannot direct commands to the managing ID. The user IDs in a managed association cannot synchronize their passwords or password phrases.

Profiles in the RRSFDATA class control whether user ID associations can be defined, to which nodes they can be defined, and which users can define them.

### RRSFDATA profiles

New RRSFDATA profiles (defined only on up-level nodes) are checked for synchronization of password phrases when a password phrase is being changed:

- ▶ The PHRASESYNC resource profile, which is similar to the PWSYNC resource profile, authorizes users for password phrase synchronization.
- ▶ The AUTODIRECT.target-node.USER.PHRSSYNC resource profile determines which password phrases get automatically directed to which nodes.

To enable synchronization of password phrases and password with RRSF, issue the SET PWSYNC command.

PWSYNC and AUTOPWD apply to password phrases as well as passwords.

### 2.5.1 Password phrase synchronization via PWSYNC

If PWSYNC is enabled between two user IDs, when the password phrase or password is changed for one of the user IDs, RACF automatically changes the password phrase or password for the other.

**Note:** Password phrase and password changes made by the RACF ADDUSER command are not synchronized because the new user ID is not part of a user ID association yet.

## Password phrase synchronization via automatic password direction

Automatic password direction (APD) extends the function of password synchronization to automatically synchronize password phrases and passwords without requiring user ID associations. The password phrase must be changed in one of the following ways in order for automatic password direction to be in effect when an application uses:

- ▶ ICHEINTY
- ▶ RACROUTE REQUEST=VERIFY
- ▶ RACROUTE REQUEST=EXTRACT, TYPE=REPLACE

Password phrase changes made in the following ways are not eligible for synchronization with automatic password direction:

- ▶ ADDUSER command
- ▶ ALTUSER command
- ▶ PASSWORD command

## 2.6 Password phrase and SETROPTS PASSWORD options

With z/OS V1R8, no new RACF SETROPTS command options are implemented; however, some of the SETROPTS PASSWORD suboperand options are new and are described as follows:

- ▶ SETROPTS PASSWORD HISTORY

The SETROPTS PASSWORD HISTORY suboperand enables you to specify the number of previous passwords and password phrases that RACF saves for each user ID and compares with an intended new value, as shown in Figure 2-13.

PASSWORD PROCESSING OPTIONS:  
12 GENERATIONS OF PREVIOUS PASSWORDS BEING MAINTAINED.

Figure 2-13 SETROPTS suboperand messages

- ▶ SETROPTS PASSWORD REVOKE

The SETROPTS PASSWORD REVOKE suboperand specifies the number of consecutive unsuccessful attempts to access the system using an incorrect password phrase or password, before RACF revokes the user ID on the next unsuccessful attempt, as shown in Figure 2-14.

**Important:** The REVOKE number specified in the SETROPTS PASSWORD applies to the combination of incorrect passwords and password phrases that RACF allows.

PASSWORD PROCESSING OPTIONS:  
AFTER 4 CONSECUTIVE UNSUCCESSFUL PASSWORD ATTEMPTS,  
A USERID WILL BE REVOKED.

Figure 2-14 SETROPTS PASSWORD REVOKE

► SETROPTS PASSWORD NOREVOKE

The SETROPTS PASSWORD NOREVOKE suboperand specifies that RACF ignores the number of consecutive unsuccessful attempts to access the system using an incorrect password phrase or password, as shown in Figure 2-15.

```
PASSWORD PROCESSING OPTIONS:
  USERIDS NOT BEING AUTOMATICALLY REVOKED.
```

Figure 2-15 SETROPTS PASSWORD NOREVOKE

► SETROPTS PASSWORD INTERVAL

The SETROPTS PASSWORD INTERVAL suboperand specifies the maximum number of days (1-254) each user ID's password phrase or password is valid, as shown in Figure 2-16.

```
PASSWORD PROCESSING OPTIONS:
  PASSWORD CHANGE INTERVAL IS 90 DAYS.
```

Figure 2-16 SETROPTS PASSWORD INTERVAL

► SETROPTS PASSWORD MINCHANGE

The SETROPTS PASSWORD MINCHANGE suboperand specifies the minimum number of days that must pass between a user ID's password phrase or password changes, as shown in Figure 2-17.

```
PASSWORD PROCESSING OPTIONS:
  PASSWORD MINIMUM CHANGE INTERVAL IS 2 DAYS.
```

Figure 2-17 SETROPTS PASSWORD MINIMUM CHANGE

► SETROPTS PASSWORD MIXEDCASE

The SETROPTS PASSWORD MIXEDCASE suboperand does not apply to password phrases because password phrases can always have lower case alphabetic characters. Figure 2-18 shows use of SETROPTS PASSWORD MIXEDCASE command.

```
PASSWORD PROCESSING OPTIONS:
  MIXED CASE PASSWORD SUPPORT IS IN EFFECT.
```

Figure 2-18 SETROPTS PASSWORD MIXEDCASE

► SETROPTS PASSWORD WARNING

The SETROPTS PASSWORD WARNING suboperand does not apply to expiring password phrases. Users are not warned of expiring password phrases. Figure 2-19 shows use of the SETROPTS PASSWORD WARNING command.

```
PASSWORD PROCESSING OPTIONS:
  NO PASSWORD EXPIRATION WARNING MESSAGES WILL BE ISSUED.
```

Figure 2-19 SETROPTS PASSWORD WARNING

► SETR PASSWORD RULE

The SETROPTS PASSWORD RULE suboperand does not apply to password phrase. Figure 2-20 shows use of the SETROPTS PASSWORD RULE command.

```
PASSWORD PROCESSING OPTIONS:  
NO INSTALLATION PASSWORD SYNTAX RULES ARE PRESENT.
```

Figure 2-20 SETROPTS PASSWORD RULE

## 2.7 Password phrase auditing

Auditing of password phrase changes is performed if the USER class is being audited, regardless of the LOGOPTIONS setting, as shown in Figure 2-21.

```
SETR LIST  
AUDIT CLASSES = USER
```

Figure 2-21 Auditing password phrase changes

## 2.8 Protected user IDs and password phrase

Protected user IDs cannot have assigned a password phrase, and cannot be revoked due to incorrect password phrase attempts or used to enter the system in ways that require a password phrase.

To protect user IDs, you need to issue the ALTUSER command with the NOPASSWORD and NOPHRASE operands, as shown in Figure 2-22.

```
ALU TEST1 NOPASSWORD NOPHRASE  
  
USER=TEST1 NAME=UNKNOWN OWNER=SYS1 CREATED=06.186  
DEFAULT-GROUP=SYS1 PASSDATE=N/A PASS-INTERVAL=N/A PHRASEDATE=N/A  
ATTRIBUTES=PROTECTED
```

Figure 2-22 ALTUSER syntax command to PROTECTED user ID.

## 2.9 Providing the ability to reset password phrases

You can allow a general user to reset user ID's password phrases using the ALTUSER command.

To provide this ability, general users need to have authority in the IRR.PASSWORD.RESET resource profile in the FACILITY class.

The following access applies to the IRR.PASSWORD.RESET resource:

► **READ access**

Users with READ access in this resource are able to use the PHRASE suboperand to change a password phrase and set it to expired for a user with an assigned password phrase who does not have the SPECIAL, OPERATIONS, or AUDITOR attribute.

**Note:** General users cannot use the password phrase operand to add a password phrase for a user who does not have one.

► **UPDATE access**

Users with UPDATE access in this resource are able to use the NOEXPIRED suboperand in conjunction with the PHRASE suboperand for a user ID who does not have the SPECIAL, OPERATIONS, or AUDITOR attribute.

► **CONTROL access**

Users with CONTROL access in this resource are able to reset a password phrase within the system's minimum change interval.

► **SPECIAL attribute**

A user who has the SPECIAL attribute can issue all RACF commands. The SPECIAL attribute gives the user full control over all of the RACF profiles in the RACF database. Users with SPECIAL access in this profile are able to use the NOEXPIRED suboperand in conjunction with the PHRASE suboperand for all users, including users who have the SPECIAL, OPERATIONS, or AUDITOR attribute.

**Note:** The SPECIAL attribute can be delegated only by a user who has the SPECIAL attribute. It should be limited to the RACF security and group administrators. Persons having the SPECIAL attribute should be required to use operator identification cards and passwords or pass phrases, and should change their passwords or pass phrases often to help ensure security.

## 2.10 RACF utilities changes

The following RACF utilities have been updated with z/OS V1R8 to reflect the password phrases:

- SMF data unload utility (IRRADU00)
- Data security monitor (DSMON)
- Database unload utility (IRRDBU00)

### 2.10.1 RACF SMF data unload utility (IRRADU00)

Prior to z/OS V1R8, the RACF SMF data unload utility (IRRADU00) is used on the JOBINIT event code qualifier 7, for revocation due to inactivity and due to invalid password attempts.

This event, 7, is now only for excessive attempts, such as excessive password or password phrase attempts.

A new event code qualifier 35, and a new text, is now used to indicate revocation of a user due to inactivity.

The explanations of the new and changed event code qualifiers follow.

- ▶ Event 1: JOB INITIATION/TSO LOGON/TSO LOGOFF:
  - (7) USER ID AUTOMATICALLY REVOKED: The user ID has been automatically revoked. The installation-defined limit of password and password phrase attempts was reached.
  - (36) PASSWORD PHRASE IS NOT VALID: A user attempted to access the system specifying a password phrase that is not valid or specifying a password phrase for a protected user ID.
  - (37) NEW PASSWORD PHRASE IS NOT VALID: Logon was rejected because new password phrase is not valid.
  - (38) CURRENT PASSWORD PHRASE HAS EXPIRED: Logon was rejected because current password phrase has expired.

## 2.10.2 RACF data security monitor (DSMON)

RACF data security monitor (DSMON) also reports the presence of the exit ICHPWX11 in the RACF exits report.

## 2.10.3 RACF database unload utility program (IRRDBU00)

There are new template fields for password phrase in the RACF database unload utility (IRRDBU00).

The new field, last password phrase date, is unloaded by IRRDBU00 as part of the user basic data record (0200):

- ▶ PHRDATE indicates the last password phrase changed date in field USBD\_PHR\_DATE.
- ▶ PHRGEN indicates the PHRASE generation number in field USBD\_PHR\_GEN.
- ▶ In addition, the USBD\_NOPWD field indicates the value “PHR” if the user is assigned a password phrase.

## 2.11 New and changed RACF messages

These are the new and changed system operator message related to the password phrases.

### New ADDUSER messages

ICH01020I PASS PHRASE REJECTED BY INSTALLATION PASSWORD PHRASE EXIT.

ICH01021I NEW PASS PHRASE REJECTED BY RACF RULES.

### New ALTUSER messages

ICH21037I PASS PHRASE CHANGE REJECTED DUE TO INSTALLATION MINIMUM CHANGE INTERVAL.

ICH21038I PASS PHRASE REJECTED BY INSTALLATION PASSWORD PHRASE EXIT.

ICH21039I NEW PASS PHRASE REJECTED BY RACF RULES.

ICH21040I PHRASE OPERAND IGNORED.

ICH21041I NOPASSWORD OPERAND IGNORED.

### **Changed ALTUSER message**

ICH21005I NOT AUTHORIZED TO SPECIFY PHRASE/NOPHRASE, OPERAND IGNORED.

### **New password phrase messages**

ICH08018I PASS PHRASE CHANGE REJECTED DUE TO INSTALLATION MINIMUM CHANGE INTERVAL.

ICH08019I PASS PHRASE CHANGE REJECTED BY INSTALLATION PASSWORD PHRASE EXIT.

ICH08020I NEW PASS PHRASE REJECTED BY RACF RULES.

ICH08021I NEW PASS PHRASE CANNOT EQUAL CURRENT PASSWORD PHRASE.

ICH08022I VALUE SPECIFIED IS NOT CURRENT PASS PHRASE.

ICH08023I PHRASE OPERAND IGNORED.

ICH08024I NEW PASS PHRASE CANNOT MATCH PREVIOUSLY USED PASSWORD PHRASE.

### **Changed password phrase message**

ICH08006I NOT AUTHORIZED TO ISSUE PHRASE.

### **New RACROUTE REQUEST=VERIFY/VERIFYX messages**

New text is added to message ICH408I:

ICH408I LOGON/JOB INITIATION - INACTIVE USER HAS BEEN REVOKED.

ICH408I LOGON/JOB INITIATION - PASS PHRASE IS NOT VALID.

### **Change RACROUTE REQUEST=VERIFY/VERIFYX message**

ICH408I LOGON/JOB INITIATION - EXCESSIVE PASSWORDS OR PASS PHRASE ATTEMPTS.

### **New RACF subsystem messages**

IRRC313I Pass phrase synchronized successfully for source-userid at source-node and target-userid at target-node.

IRRC318I Unable to set pass phrase date. Return code is retun-code. Reason code is reason-code.

IRRC320I Phrase date was set for userid at node-name.



## Availability improvements for IRRUT200 and IRRUT400

The zSeries® mainframe systems are well known to provide the highest degree of availability. Security and availability correlate and are both important systems management disciplines. In this chapter availability as a subtheme of security is described. In addition to the IBM Health Checker for z/OS enhancements, there are further availability improvements in z/OS V1R8.

In this chapter, the new RACF availability updates for the database utilities IRRUT200 and IRRUT400 are described.

The significant improvements that have been made to the two utilities are:

- ▶ Synchronized copy of RACF data base with IRRUT200
- ▶ Safety features for IRRUT200 and IRRU400
- ▶ Dependencies and migration considerations
- ▶ Updates to publications

## 3.1 Synchronized copy with IRRUT200

The IRRUT200 utility existed prior to z/OS V1R8 and had the following two major functions:

- ▶ Verification of the RACF database to identify possible inconsistencies in the internal organization, as follows:
  - Against just the SYSRACF DD data set, which runs with serialization
  - Against just the SYSUT1 DD data set, which runs without serialization
- ▶ Creating a copy of the specified input database, as follows:
  - Against both the SYSRACF DD and the SYSUT1 DD, the SYSRACF DD data set is copied under serialization to the SYSUT1 DD data set. Then SYSUT1 is verified unserialized.

The new *synchronized* copy now allows a serialized copy of an in-use primary database to its corresponding inactive backup. The result is an active backup whose synchronization with the primary database is guaranteed by serialization.

Prior to this feature you had to organize the backup and re-synchronization process of the RACF database. This feature simplifies those operations.

### 3.1.1 Pre-z/OS V1R8 implementation

There might be several situations in which you have the challenge to create an exact copy of your primary database, for example, if you want to relocate a RACF data set to another volume.

In performing this task prior to z/OS V1R8, the whole task was a multi-step procedure and there was the possibility of it not being well synchronized.

#### Steps to copy database

Prior to z/OS V1R8, the following steps could be used:

1. Run the IRRUT200 utility to copy the active primary to a data set with the same name as the active backup, but on a different volume. During this copy the utility does a verification to be sure that the primary database is not damaged.
2. If the utility runs without error, the RVARY INACT command is used to inactivate the backup data set.
3. Uncatalog the old backup data set.
4. Catalog the new copy.
5. Use the RVARY ACTIVE command to activate the new copy as the RACF backup database.

**Important:** If updates to the primary RACF data base occur between the IRRUT200 copy function (step 1) and the activation of the new copy (step 5), then the backup is no longer an exact copy of the primary. The new synchronized copy solves this problem.

### 3.1.2 Synchronized copy solution

The IRRUT200 utility is modified and introduces an additional PARM=ACTIVATE parameter. With this parameter you can create a synchronized copy in the same way as some of the steps previously mentioned. Some of those steps are now done in the background and are

serialized. You no longer have to do the activation with the RVARY ACTIVE command. It is done internally by the IRRUT200 utility and thus makes the process easier and more consistent.

The ACTIVATE parameter can ensure that no updates are made to the input data set between the time that it is copied and the time that the copy is activated. However, it can only ensure a synchronized copy if the system on which the utility is running is in RACF sysplex communications mode, or the RACF data set is not shared with another system. If other systems share the backup data set and are not in sysplex communications mode, IRRUT200 can only activate the data set on the system on which the utility is running. To activate the backup data set on the sharing systems, you must issue an RVARY ACTIVE command.

## Using synchronized copy

The following steps describe the new process of a synchronized copy:

1. Verify that the primary database is not damaged by running the IRRUT200 utility in verification mode.

**Note:** This first step is necessary because the synchronized copy does no verification, and if your primary RACF is damaged, as a result of the synchronized copy you will have two in-use damaged RACF data sets.

Figure 3-1 shows the IRRUT200 utility in verify mode.

```
//VERIFY JOB
//STEP EXEC PGM=IRRUT200
//SYSRACF DD DSN=SYS1.RACFESA,DISP=SHR
//SYSUT1 DD UNIT=SYSDA,SPACE=(CYL,(10)),
// DCB=(LRECL=4096,RECFM=F)
//SYSUT2 DD SYSOUT=A
//SYSPRINT DD SYSOUT=A
//SYSIN DD
INDEX
MAP
END
/*
```

Figure 3-1 JCL for IRRUT200 utility in verify mode

2. If the utility runs without any errors, use the RVARY INACT command to inactivate the backup data set. A RVARY LIST command at this time should give the result shown in Figure 3-2.

```
RVARY LIST
IRRA011I (#) OUTPUT FROM RVARY: 366
ICH15013I RACF DATABASE STATUS:
ACTIVE USE  NUM VOLUME  DATASET
-----
YES  PRIM   1 BH5CAT   SYS1.RACFESA
NO   BACK   1 *DEALLOC SYS1.RACFBK
ICH15020I RVARY COMMAND HAS FINISHED PROCESSING.
```

Figure 3-2 RVARY list output after inact of the backup data set

3. Run the IRRUT200 utility with the new keyword PARM=ACTIVATE as shown in Figure 3-3. IRRUT200 synchronizes the primary and the backup RACF data sets. After successful completion, the backup data set is an exact copy of the primary and both data sets will be active.

```
//SYNCPY JOB
//COPYDS EXEC PGM=IRRUT200,PARM='ACTIVATE'
//SYSPRINT DD SYSOUT=*
//SYSRACF DD DSN=SYS1.RACFESA,DISP=SHR
//SYSUT1 DD DSN=SYS1.RACFBK,DISP=OLD
//SYSUT2 DD SYSOUT=*
//SYSIN DD *
```

Figure 3-3 JCL for synchronized copy with IRRUT200

The SYSUT2 output data set contains the result of the synchronized copy as shown in Figure 3-4.

```
DATA SET UTILITY - GENERATE

PROCESSING ENDED AT EOD
IRR62065I - IEBGENER copied SYSRACF to the work dataset SYSUT1, IEBGENER
RC=0000
ICH15013I RACF DATABASE STATUS:
ACTIVE USE  NUM VOLUME  DATASET
-----
YES  PRIM   1 BH5CAT   SYS1.RACFESA
YES  BACK   1 BH5CAT   SYS1.RACFBK
ICH15020I RVARV COMMAND HAS FINISHED PROCESSING.
IRR62070I Backup data set activated on this system only. Synchronization of
primary and backup cannot be guaranteed.
IRR62071I SYSIN DD statement ignored. PARM=ACTIVATE has been specified.
```

Figure 3-4 Output of the synchronized copy

**Attention:** The IRR62070I message is an informational message and has importance if you share your RACF data sets in a sysplex without sysplex communication mode enabled. In this case it is necessary to issue the RVARV ACTIVATE command on the other systems. In sysplex communication mode the RVARV command is propagated.

The IRR62070I message appears if the DASD upon which the backup data set resides is generated SHARED and if the system is not in sysplex communication mode.

### Details on the synchronized copy

If you invoke the IRRUT200 utility with the PARM=ACTIVATE keyword, you should be aware of the following things:

- ▶ PARM=ACTIVATE does no verification; SYSPRINT and SYSIN DD statements are ignored.
- ▶ SYSUT1 is the target of the output and must be the in-use inactive backup associated with the SYSRACF (source of input) in-use active primary as shown in Figure 3-2 on page 23.
- ▶ IRRUT200 gets exclusive serialization.

- ▶ When the copy is complete, an internal RVARY ACTIVE command is done against the SYSUT1 (target of output) backup and the serialization is released.
- ▶ Besides the IRRUT200 messages, SYSUT2 contains the RVARY messages.

**Attention:** You are not prompted for an RVARY password. Thus the IRRUT200 utility has become more powerful and you should restrict its use with RACF PROGRAM control.

- ▶ The existing IRRUT200 rule of copying only between similar devices still applies.

### 3.1.3 Interaction, dependencies, and migration considerations

The synchronized copy feature has no hardware requirements but requires at least z/OS V1R8 and will not be rolled back to older z/OS releases.

## 3.2 Safety features for IRRUT200 and IRRUT400

The utilities IRRUT200 and IRRUT400 existed prior to z/OS V1R8 and both can be used to copy a RACF data set.

The main differences are as follows:

- ▶ IRRUT200 copies or verifies on an 1:1 basis, meaning the target and source have the same sizes and device characteristics.
- ▶ IRRUT400 copies on a n:m basis, meaning that the utility can be used to expand the database from n to n+m data sets, or decrease it from n to n-m data sets. The data sets may reside on different device types and be of different sizes.

The new safety feature of both utilities protects against instances where an improper invocation would have led to RACF database corruption in the past and thus affect your system availability.

### Pre-z/OS V1R8 support

If these utilities were used in the past to copy your data set, that data set could possibly unintentionally corrupt your output data set.

### IRRUT200

The utility IRRUT200 has two important DD names. When both are specified and a copy is to be made from the SYSRACF DD data set to the SYSUT1 DD data set, as follows, then corruption would occur:

<b>SYSRACF</b>	Refers to the <i>source</i> (or <i>input</i> ) data set from which the copy is taken.
<b>SYSUT1</b>	Specifies to the <i>target</i> (or <i>output</i> ) data set.

There are two scenarios which occurred in the past and damaged a RACF data base:

1. If you specified the *same* data set name for SYSRACF (source/input) and the SYSUT1 (target/output), then corruption of the data base might occur.
2. Corruption might also occur if SYSUT1 points to an *active* in-use data set.

## IRRUT400

The IRRUT400 utility works with the following DD names to describe source and target data sets:

**INDDn** Refers to the *source* or *input* data sets from which the copy is taken.

**OUTDDn** Specifies the *target* or *output* data sets.

Data base corruption might occur if:

1. You specified the *same* data set name for INDDn (source/input) and OUTDDn.
2. OUTDDn is an *active* in-use data set.

### 3.2.1 Safety features with z/OS V1R8

The new safety features for both utilities are very similar. New checks which take volume labels into account are added. The utilities no longer allow use of the same data set for *target* and *source* or the use of an active in-use data set as a *target*.

There are no changes to the JCL because whenever you invoke IRRUT200 and IRRUT400 the safety features are integrated.

### 3.2.2 Safety feature implementation examples

Even though the concept is the same for IRRUT200 and IRRUT400, there are different messages for the scenarios.

#### IRRUT200

When the DD names SYSRACF(source) and SYSUT1(target) point to the same data sets, as shown in Figure 3-5, the utility now terminates with a RC=12 and an IRR62073I message is in the output in DD name SYSUT2, as shown in Figure 3-6.

```
//COPYSAME JOB (POK,999),MSGCLASS=T,NOTIFY=&SYSUID
//STEP01 EXEC PGM=IRRUT200
//SYSPRINT DD SYSOUT=*
//SYSRACF DD DSN=PLEUS.SYS1.RACFESA,DISP=SHR
//SYSUT1 DD DSN=PLEUS.SYS1.RACFESA,DISP=SHR
//SYSUT2 DD SYSOUT=*
//SYSIN DD *
INDEX
MAP
END
/*
```

Figure 3-5 Sample JCL for IRRUT200 with same DD names for SYSRACF and SYSUT1

```
IRR62073I Database copy (SYSRACF to SYSUT1) failed. Same data set specified
for input and output.
```

Figure 3-6 SYSUT2 output for IRRUT200 with same DD names for SYSRACF and SYSUT1

When both DD names SYSRACF(source) and SYSUT1 are specified (and point to different data set names), but SYSUT1(target) is an *active* RACF data set, the utility terminates with RC=12 and IRR62072I message.

The RVARY LIST command shows you the status of your RACF data sets, as shown in Figure 3-7.

```
#RVARY LIST
IRRA011I (#) OUTPUT FROM RVARY: 442
ICH15013I RACF DATABASE STATUS:
ACTIVE USE  NUM VOLUME  DATASET
-----
YES  PRIM   1 BH5CAT   SYS1.RACFESA
YES  BACK   1 BH5CAT   SYS1.RACFBK
ICH15020I RVARY COMMAND HAS FINISHED PROCESSING.
```

Figure 3-7 Output of RVARY LIST command.

```
//COPYACT JOB (POK,999),MSGCLASS=T,NOTIFY=&SYSUID
//STEP01 EXEC PGM=IRRUT200
//SYSRACF DD SYSOUT=*
//SYSRACF DD DSN=SYS1.RACFESA,DISP=SHR
//SYSUT1 DD DSN=SYS1.RACFBK,DISP=SHR
//SYSUT2 DD SYSOUT=*
//SYSIN DD *
INDEX
MAP
END
/*
```

Figure 3-8 JCL for IRRUT200 and active SYSUT1

If you start the JCL in Figure 3-8 on a system with the new safety feature, you receive the output in DD name SYSUT2 shown in Figure 3-9.

```
IRR62072I Database copy (SYSRACF to SYSUT1) failed. SYSUT1 is an active RACF
data set.
```

Figure 3-9 SYSUT2 output from IRRUT200 with active SYSUT1

## IRRUT400

When the DD names INDDn(source) and OUTDDn(target) point to the same data set, the utility will terminate with RC=16 and message IRR6504I.

```
//COPYSAME JOB (POK,999),MSGCLASS=T,NOTIFY=&SYSUID
//STEP01 EXEC PGM=IRRUT400,PARM='NOLOCKINPUT,FREESPACE(20)'
//SYSRACF DD SYSOUT=*
//INDD1 DD DSN=PLEUS.SYS1.RACFESA,DISP=SHR
//OUTDD1 DD DSN=PLEUS.SYS1.RACFESA,DISP=SHR
/*
```

Figure 3-10 Sample JCL for IRRUT400 with same DD names for INDD1 and OUTDD1

If you run the JCL in Figure 3-10 on a system with the new safety feature, you get the output in DD name SYSRACF shown in Figure 3-11 on page 28.

```

IRR65020I Specified Options: NOLOCKINPUT,FREESPACE(20)
IRR65025I Options in Effect:
NOALIGN,NODUPDATASETS,NOTABLE,FREESPACE(20),NOLOCKINPUT
IRR65008I PLEUS.SYS1.RACFESA successfully opened for processing on INDD1 .
IRR65041I Database copy (INDD1 to OUTDD1) failed. Same dataset specified for
input and output.

```

Figure 3-11 SYSPRINT output for IRRUT400 with same DD names for INDD1 and OUTDD1

When both DD names INDDn(source) and OUTDDn are specified (and point to different data sets), but the OUTDDn(target) is an *active* RACF data set, the utility terminates with RC=16 and message IRR65040I.

To find out what your active RACF data sets are, you can use the RVARY LIST command as shown in Figure 3-7 on page 27.

```

//COPYACT JOB (POK,999),MSGCLASS=T,NOTIFY=&SYSUID
//STEP01 EXEC PGM=IRRUT400,PARM='NOLOCKINPUT,FREESPACE(20)'
//SYSPRINT DD SYSOUT=*
//INDD1 DD DSN=SYS1.RACFESA,DISP=SHR
//OUTDD1 DD DSN=SYS1.RACFBK,DISP=SHR
/*
//

```

Figure 3-12 JCL for IRRUT400 and active OUTDD1

If you submit the JCL in Figure 3-12 on a system with the new safety feature, you get the output in DD name SYSPRINT shown in Figure 3-13.

```

IRR65020I Specified Options: NOLOCKINPUT,FREESPACE(20)
IRR65025I Options in Effect:
NOALIGN,NODUPDATASETS,NOTABLE,FREESPACE(20),NOLOCKINPUT
IRR65008I SYS1.RACFESA successfully opened for processing on INDD1 .
IRR65040I Output processing failed. OUTDD1 Specifies an active RACF database on
this system.

```

Figure 3-13 SYSPRINT output from IRRUT400 with an active OUTDDx

### 3.2.3 Interaction, dependencies, and migrations considerations

The safety features for the utilities IRRUT200 and IRRUT400 are available with z/OS V1R8 and via a small programming enhancement (SPE) in z/OS V1R7. You have to install APAR OA14916 to use it in z/OS V1R7.

## 3.3 Publication updates

These two new features, the synchronized copy and the safety features for IRRUT200 and IRRUT400, come along with new and updated messages and publications.



### 3.3.1 Publications

The following publications reflect these new features:

- ▶ *z/OS Security Server RACF System Programmers Guide, SA22-7681*  
Updated for IRRUT200 PARM=ACTIVATE and minor changes for the safety features for IRRUT200 and IRRUT400.
- ▶ *z/OS Security Server RACF Messages and Codes, SA22-7678*  
Updated for IRR417I and all new IRRUT200 and IRRUT400 messages.

### 3.3.2 Changed IRRUT200 messages

This is an updated message and the new reason for getting this message would be a failure during an IRRUT200 PARM=ACTIVATE attempt.

**IRR417I** UNABLE TO COMMUNICATE WITH THE RACF SUBSYSTEM. IEFSSREQ RETURN-CODE IS *return\_code*.

### 3.3.3 New IRRUT200 messages

Following are the new IRRUT200 messages:

**IRR62069I** Database copy and activation failed. SYSUT1 does not identify the inactive backup data set for SYSRACF.

**IRR62070I** Backup data set activated on this system only. Synchronization of primary and backup cannot be guaranteed.

**IRR62071I** SYSIN DD statement ignored. PARM=ACTIVATE has been specified.

**IRR62072I** Database copy (SYSRACF to SYSUT1) failed. SYSUT1 is an active RACF data set.

**IRR62073I** Database copy (SYSRACF to SYSUT1) failed. Same data set specified for input and output.

**IRR62074I** RVARY ACTIVE failure against SYSUT1, RC = *return-code*

**IRR62075I** *service* on *ddname* failure, RC = *return-code*

**IRR62076I** Parameter error. Text beginning with ' ' contains an undefined keyword.

### 3.3.4 New and changed IRRUT400 messages

Following are new and changed messages for IRRUT400:

**IRR65040I** Output processing failed. *ddname* specifies an active RACF database on this system.

**IRR65041I** Database copy (INDDx to OUTDDx) failed. Same data set specified for input and output.

Archived

## RACF and the DB2 access control module

In this chapter, the RACF methods are described that take advantage of the new RACF access control module exit, DSNXRXAC, which is supplied with DB2 V9. The DSNXRXAC exit is a replacement for the RACF IRR@XACS exit that was supplied for use in earlier DB2 releases.

The RACF access control module has changed substantially with DB2 Version 9. It continues to integrate DB2 processing with RACF security, allowing for the consolidation of security administrative tasks and audit logs by moving the security management of DB2 objects and users into the RACF database. A RACF administrator can define a security rule before an object is created and preserve the rule for a dropped object. In addition, RACF general resources for member/grouping profiles can be utilized by an installation to protect multiple DB2 resources with a single RACF profile; RACF's group authorization mechanism is simpler to use than DB2's "secondary auth ID."

DB2 Version 9 also offers:

- ▶ New RACF general resource class MDSNSQ for SEQUENCE objects in the class descriptor table (CDT) ICHRRCDX.
- ▶ DB2 V9 has extended the lengths of many of its constructs and several RACF classes have been updated to support a maximum profile length of 246 characters.
- ▶ New WARNING mode support as an enhancement to the existing support in IRR@XACS.
- ▶ Multilevel security with row-level granularity that restricts access to an object or row based on the security label of the object or row and the security label of the user. This is unrelated to the DSNXRXAC exit.

## 4.1 Previous DB2 versions

Prior to DB2 Version 5 and OS/390 V2R4, only the DB2 native security mechanisms of GRANT and REVOKE could be used to control access to DB2 objects such as tables, views, and databases. DB2 administrators were also required to be security administrators, and in an effort to centralize the security administrative tasks, a dummy exit point was introduced with DB2 V5 that offered the ability to move some of the administrative tasks to RACF.

In order to take advantage of RACF, it was necessary to replace this dummy exit point with the RACF DB2 external security module, which was provided as a sample assembler language routine in SYS1.SAMPLIB(IRR@XACS). Once the exit was installed it was called whenever access control decisions were needed for integrated DB2 processing with RACF security.

### DB2 V9

The IRR@XACS exit is no longer shipped with RACF in SYS1.SAMPLIB. It has changed substantially and is now shipped with DB2 as FMID HDRE810. As a result of this the previous exit is no longer usable with DB2 V9. If you have the RACF/DB2 external security module installed it will be necessary to migrate to the new RACF access control module for DB2 V9.

### 4.1.1 Security implementation

Implementing the RACF access control module involves the interaction of RACF, DB2, and z/OS system software, each with its own required skills. It is important to get the correct system programmers together for the planning and implementation of the RACF access control module.

This involves the following planning decisions.

1. There are two types of migrations to consider:
  - a. Migrating from DB2 internal security where you do not use RACF for access control authorization for DB2 resources
  - b. Migrating from a previous level of the RACF and DB2 external security module where you are already using RACF for access control authorization to DB2 resource
2. There are two approaches regarding the RACF access control module: using IBM defaults or customizing the values.

- a. One approach is to use the IBM default values. This enables you to place profiles for all DB2 subsystems into one set of RACF general resource classes.

Using one set of RACF general resource classes offers several advantages:

- You can use the IBM-supplied general resource classes.
- You can consolidate all of your DB2 access control for all DB2 subsystems into one place.
- You can use the RACF GENERIC OWNER facility to delegate the administration of DB2 access control at the subsystem, database, tablespace, table, or even the view level.

- b. The other approach is to customize the RACF access control module and use separate sets of RACF general resource classes for each DB2 subsystem. The benefits this provides are:
    - Isolation of profiles from differing DB2 subsystems
    - Fewer profiles in each general resource class

**Note:** When choosing to use RACF for DB2 administrative tasks for the first time, it is not necessary to migrate protection of all DB2 objects at once. If the RACF access control module cannot find a RACF profile to protect a particular object, it defers to DB2 authority checking.

### 4.1.2 Expanding RACF protection to DB2 objects

The most significant part of the planning is to expand RACF administration to the DB2 subsystem and its resources. Include the following tasks in your planning process:

1. Examine the current RACF environment, including user group structure, resource naming conventions, and the use of grouping classes.
2. Examine the DB2 objects. Look for naming conventions and other similarities in resource names that can be exploited with RACF generic profiles.
3. Examine the GRANT authorizations in place for DB2 objects to see which RACF user groups you can define or exploit.
4. Plan which DB2 objects and administrative authorities to protect, determine access requirements, and incorporate the new subsystem into the RACF structure.
5. Consider using RACF variables to facilitate resource naming conventions for DB2 resources.
6. Integrate new DB2 users into the RACF user structure using the RACF group and class authorities.

Refer to the planning section of *DB2 UDB for z/OS V8 RACF Access Control Module Guide*, SC18-7433 for details.

## 4.2 The RACF access control module - DSNXRXAC

The RACF access control module allows you to use RACF in addition to DB2 authorization checking for DB2 objects, authorities, commands, and utilities. You can activate the RACF access control module at the DB2 access control authorization exit point (DSNX@XAC), where you can replace the default DB2 exit routine.

The RACF access control module requires DB2 V8 or later, and is supplied as an assembler source module in the DSNXRXAC member of prefix.SDSNSAMP. Support for earlier DB2 versions was supplied in the IRR@XACS member of SYS1.SAMPLIB and was called the DB2 RACF external security module.

The RACF access control module does the following:

- Receives control from the DB2 access control authorization exit point, DSNX@XAC, to handle DB2 authorization checks
- Provides a single point of control for RACF and DB2 security administration
- Provides flexibility for multiple DB2 subsystems with a single set of RACF profiles
- Allows you to validate a user ID before giving it access to a DB2 object

RACF supplies a security exit, which enables SAF/RACF services to be utilized for DB2 security, effectively allowing DB2 security to be centralized within the scope of the system security administrator. The following changes have been implemented:

1. When a security decision needs to be made, DB2 builds a control block, DSNDXAPL, that contains security information and passes it to the common security module DSNX@XAC.
2. The DSNX@XAC exit uses information in the control block XAPL and the rules tables to construct a series of RACROUTE FASTAUTH requests.
3. The result of the RACROUTE FASTAUTH request is passed as a return code to the DB2 resource manager, as shown in Figure 4-1. The return codes have these meanings:

<b>Return code 0</b>	Allow access.
<b>Return code 8</b>	Do not allow access.
<b>Return code 4</b>	No profile defined in RACF for this object. DB2 security will make the final decision.

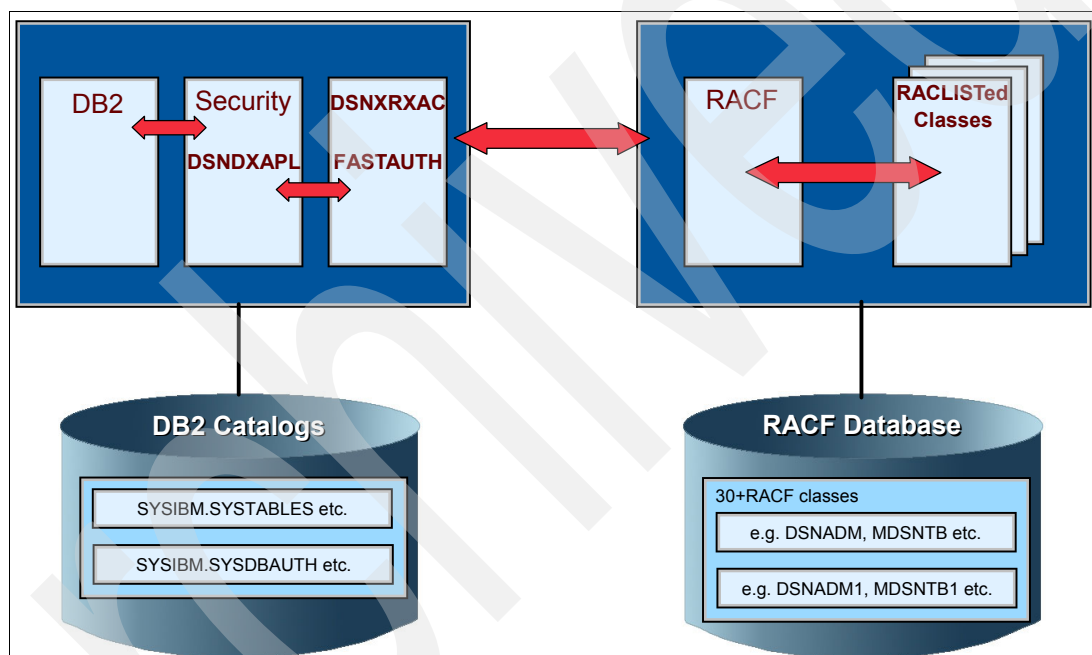


Figure 4-1 Diagram of DSNRXAC processing

**Note:** The exit point associated with the RACF access control module is a DB2 exit point, not a RACF exit point. DB2 provides a dummy exit as the default. If you want to use RACF authorization for DB2 objects you need to replace the dummy exit with the RACF access control module (DSNRXAC).

### DSNRXAC advantages

There are many advantages to using the RACF access control module to control authorization to DB2 resources from RACF. DSNRXAC provides the ability to:

- Validate auth IDs before DB2 authorities are granted.
- Define security rules before the object is created.
- Preserve security rules for dropped objects.
- Preserve DB2 privileges and administrative authorities.

- ▶ Have flexibility for multiple DB2 subsystems with a single set of RACF profiles.
- ▶ Protect multiple DB2 objects with a single security rule using a combination of RACF generic, grouping, and member profiles.
- ▶ Control and audit resources for multiple DB2 subsystems from a single point.
- ▶ Administer DB2 security with a minimum of DB2 skills.
- ▶ Eliminate DB2 cascading revoke.

The RACF access control module is invoked:

- ▶ Once at DB2 subsystem startup to perform any required setup prior to authorization checking. Authorization checking profiles are also loaded at this point.
- ▶ For each DB2 subsystem authorization request.
- ▶ Once when the DB2 subsystem terminates to perform its cleanup before DB2 stops.

### 4.2.1 Modifying the RACF access control module

The RACF access control module is shipped in *prefix.SDSNSAMP.DB2*. DB2 V9 provides two sets of the RACF access control module source:

- DSNXSXAC**      Dummy default version. A compiled version is present in SDSNLOAD for use when you are not using external security.
- DSNXRXAC**      Sample version that allows the use of RACF for external control of DB2 resources. This version can be modified if desired.

**Note:** You only need to modify DSNRXAC if you are not planning on using the default RACF general resource classes that are supplied in the class descriptor table (CDT). The values in the SET symbols located at the top of the source dataset can be altered and the module recompiled and linked into the *prefix.SDSNLOAD* data set.

Table 4-1 lists some of the variables that you can alter to customize the DSNRXAC module. This process is described in detail in the *z/OS Security Server RACF Systems Programmer's Guide*, SA22-7681.

Table 4-1 RACF access control module assembler SET options

SET symbol	Default value	Description
&CLASSOPT	2	Specifies the classification option to be used: 1 = single subsystem scope 2 = multiple subsystem scope
&CLASSMNT	DSN	Specifies the classname root. Characters 2-5 of the class name. It is only valid for &CLASSOPT=2.
&CHAROPT	1	Specifies the one character name suffix to be used if you are not using the RACF default class names.

**Restriction:** Each SET option chosen applies to the DB2 subsystem using the DSNRXAC module. If you have different DB2 subsystems that require different SET options, the RACF access control module will have to be installed separately for each subsystem, each with its own SET options.

## 4.2.2 Activating the RACF access control module

The following steps are mandatory and must be completed whether you have chosen to customize the RACF access control module or to use the IBM default class names.

1. Copy *prefix.SDSNSAMP*(DSNXRXAC) to a private library as *private.library*. DSNX@XAC.  
Optionally, at this point, customize your private copy of the RACF access control module by modifying the assembler SET options.
2. Use the DB2 installation job DSNTIJEX to assemble and linkedit the module into the APF authorized DB2 exit load library (*prefix.SDSNEXIT*). The exit routine must have a CSECT name and an entry point with the same name DSNX@XAC.
3. Once you have installed the RACF access control module it will become active the next time the DB2 subsystem is restarted when at least one RACF class associated with the DB2 system is active.

For a detailed description of this process refer to *DB2 Universal Database for z/OS RACF Access Control Module Guide*, SC18-7433.

Figure 4-2 on page 37 is sample JCL to assemble and linkedit the module RACF access control module. This JCL can be found in *prefix.SDSNSAMP*.



```

//ASMPROC PROC WSPC=500, MEM=TEMPNAME, MEM1=TEMPNAME
//*
//*          ASSEMBLE
//*
//ASM      EXEC PGM=ASMA90, PARM=' OBJECT, NODECK '
//SYSIN    DD DISP=SHR,
//          DSN=DB8Y8.SDSNSAMP(&MEM)
//SYSLIB   DD DSN=SYS1.MODGEN, DISP=SHR
//          DD DSN=SYS1.MACLIB, DISP=SHR
//          DD DISP=SHR,
//          DSN=DB8Y8.SDSNMACS
//          DD DISP=SHR,
//          DSN=DB8Y8.SDSNSAMP
//SYSLIN   DD DSN=&&LOADSET, DISP=(MOD,PASS), UNIT=VIO,
//          SPACE=(800,(&WSPC,&WSPC)), DCB=(BLKSIZE=800)
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSUT1   DD UNIT=VIO, SPACE=(800,(&WSPC,&WSPC)),, ROUND)
//SYSUT2   DD UNIT=VIO, SPACE=(800,(&WSPC,&WSPC)),, ROUND)
//SYSUT3   DD UNIT=VIO, SPACE=(800,(&WSPC,&WSPC)),, ROUND)
//*
//LKED     EXEC PGM=IEWL, REGION=1024K,
// PARM=' SIZE=(900K,124K), LIST, XREF, RENT, OL, NCAL,
// AMODE=31, RMODE=ANY',
// COND=(4,LT,ASM)
//SYSLIN   DD DSN=&&LOADSET, DISP=(OLD,DELETE)
//DSNLOAD  DD DSN=DB8Y8.SDSNLOAD, DISP=SHR
//SYSLMOD  DD DISP=SHR,
//          DSN=DB8Y8.SDSNEXIT(&MEM1)
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSUT1   DD UNIT=VIO, SPACE=(1024,(50,50))
//ASMPROC  PEND
//*****
//* STEP 3 : ASSEMBLE AND LINK EDIT ACCESS CONTROL
//*          AUTHORIZATION EXIT AND REPLACE
//*          DSNXSXAC WITH YOUR ACA EXIT ROUTINE NAME
//*****
//JEX0003 EXEC ASMPROC, MEM1=DSNX@XAC, MEM=DSNXRXAC
//LKED.SYSLIN DD
//          DD *
//          ENTRY DSNX@XAC
//          NAME DSNX@XAC(R)

```

Figure 4-2 Sample JCL to assemble and linkedit the RACF access control module

### Activating the RACF/DB2 resource classes

You must have at least one RACF/DB2 class active and RACLISTed with an associated resource profile at DB2 subsystem startup in order to activate the RACF access control module. Figure 4-3 on page 38 shows a batch job that issues a SETROPTS command to achieve this.

```
//STPA EXEC PGM=IKJEFT01,DYNAMNBR=20
//SYSTSPRT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSTSIN DD *
SETROPTS CLASSACT(MDSNTB) RACLIST(MDSNTB)
```

Figure 4-3 Activating a DB2 resource class

**Note:** Remember it is only necessary to activate one class and profile to activate the exit. However, to add another resource class will necessitate a recycling of DB2. Updates to profiles, on the other hand, will only require a SETR RACLIST REFRESH of the class to activate.

## RACF resource classes and DB2 objects

The relationship between the RACF resource classes and DB2 objects is listed in Table 4-2. You must plan in advance which DB2 objects you wish to initially protect because in order to introduce a new DB2 resource it will be necessary to stop and restart DB2. DSNR is a DB2-related class (connecting to subsys name) but is unrelated to the exit.

Table 4-2 RACF resource classes and their corresponding DB2 objects

Class name	Description
DSNADM	DB2 administrative authority class
DSNR	Control access to DB2 subsystems
GDSNBP	Grouping class for DB2 buffer pool privileges
GDSNCL	Grouping class for DB2 collection privileges
GDSNDB	Grouping class for DB2 database privileges
GDSNJR	Grouping class for Java™ archive files (JARS)
GDSNPK	Grouping class for DB2 package privileges
GDSNPN	Grouping class for DB2 plan privileges
GDSNSC	Grouping class for DB2 schemas privileges
GDSNSG	Grouping class for DB2 storage group privileges
GDSNSM	Grouping class for DB2 system privileges
GDSNSP	Grouping class for DB2 stored procedures
GDSNSQ	Grouping class for DB2 sequences
GDSNTB	Grouping class for DB2 table, index or view privileges
GDSNTS	Grouping class for DB2 tablespace privileges
GDSNUF	Grouping class for DB2 user-defined function privileges
GDSNUT	Grouping class for DB2 user-defined distinct type privileges
MDSNBP	Member class for DB2 buffer pool privileges
MDSNCL	Member class for DB2 collection privileges

Class name	Description
MDSNDB	Member class for DB2 database privileges
MDSNJR	Member class for Java archive files (JARS)
MDSNPK	Member class for DB2 package privileges
MDSNPN	Member class for DB2 plan privileges
MDSNSC	Member class for DB2 schema privileges
MDSNSG	Member class for DB2 storage group privileges
MDSNSM	Member class for DB2 system privileges
MDSNSP	Member class for DB2 stored procedures privileges
MDSNSQ	Member class for DB2 sequences
MDSNTB	Member class for DB2 table, index or view privileges
MDSNTS	Member class for DB2 tablespace privileges
MDSNUF	Member class for DB2 user-defined function privileges
MDSNUT	Member class for DB2 user-defined distinct type privileges

Figure 4-4 shows sample JCL that adds a generic profile to the MDSNTB resource class in RACF and performs a SETROPTS RACLIST REFRESH to activate that profile.

```
//STEPS EXEC PGM=IKJEFT01,DYNAMNBR=20
//SYSTSPRT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSTSIN DD *
RDEF MDSNTB ** UACC(NONE)
SETROPTS RACLIST(MDSNTB) REFRESH
```

Figure 4-4 Activating and RACLISTing the RACF/DB2 resource classes

### 4.2.3 Restarting DB2 with the RACF access control module

Once the RACF access control module has been installed and the RACF classes being used are active, you must stop and start DB2 to invoke the RACF access control module.

From the MVS console issue the following commands:

```
-STOP DB2
-START DB2
```

Once the DB2 subsystem has been initialized, issue a RACLIST refresh to activate any profile changes to classes that are being used by the RACF access control module.

Figure 4-5 on page 40 shows sample JCL for a new profile that has been added to the MDSNTB resource class with an RDEFINE command. This is followed by a PERMIT command that will authorize USER01 to access the database. Finally, the resource class is refreshed in storage to activate the change with a SETROPTS RACLIST REFRESH command.

When migrating from DB2 internal security to the RACF access control module consider using the WARNING option to RDEFINE and RALTER commands to protect DB2 objects. RACF will issue an ICH408I message if a user ID has insufficient authority to access that profile but will still allow access.

Figure 4-5 is a sample job that will issue an RDEFINE to create a resource profile for a table in class (MDSNTB). It also issues a PERMIT command to allow USER01 to access this profile and finally issues the SETROPTS to activate the new profile.

**Note:** If the WARN option is added to a resource that is requested by a user with a DB2 administrative authority, such as SYSADM, DBADM, or in some cases SYSCTRL, which normally allows the user to access the object, the user can ignore the warning messages.

```
//STPA EXEC PGM=IKJEFT01,DYNAMNR=20
//SYSTSPRT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSTSIN DD *
RDEF CLASS(MDSNTB) DB2.SYSIBM.SYSDATABASE.ALTER UACC(NONE)
PE DB2.SYSIBM.SYSDATABASE.ALTER ID(USER01) ACCESS(READ) CLASS(MDSNTB)
SETROPTS RACLIST(MDSNTB) REFRESH
```

Figure 4-5 Activating a RACF profile change - equivalent to a DB2 GRANT on a table

**Restriction:** The only way to add another class that was not already ACTIVE and RACLISTED at DB2 startup time is to issue SETROPTS CLASSACT(CLASS) RACLIST and then to recycle DB2.

Figure 4-6 on page 41 shows the messages that you will see in the DB2 log to indicate that the RACF access control module has been successfully invoked at DB2 startup.

```

IRR908I RACF/DB2 EXTERNAL SECURITY MODULE FOR DB2 SUBSYSTEM DB8Y HAS 329
        A MODULE VERSION OF HDRE810  AND A MODULE LENGTH OF 00005BD0.
IRR909I RACF/DB2 EXTERNAL SECURITY MODULE FOR DB2 SUBSYSTEM DB8Y 330
        IS USING OPTIONS: &CLASSOPT=2
                        &CLASSNMT=DSN
                        &CHAROPT=1
                        &ERROROPT=1
                        &PCCELLCT=50
                        &SCCELLCT=50
IRR910I RACF/DB2 EXTERNAL SECURITY MODULE FOR DB2 SUBSYSTEM DB8Y 331
        INITIATED RACLIST FOR CLASSES:
        MDSNDB  MDSNPK  MDSNPN  MDSNBP  MDSNCL
        MDSNTS  MDSNSG  MDSNTB  MDSNSM  MDSNSC
        MDSNUT  MDSNUF  MDSNSP  MDSNJR  MDSNSQ
        DSNADM
IRR911I RACF/DB2 EXTERNAL SECURITY MODULE FOR DB2 SUBSYSTEM DB8Y 332
        SUCCESSFULLY RACLISTED CLASSES:
        MDSNDB  MDSNPK  MDSNPN  MDSNBP  MDSNCL
        MDSNTS  MDSNSG  MDSNTB  MDSNSM  MDSNSC
        MDSNUT  MDSNUF  MDSNSP  MDSNJR  MDSNSQ
        DSNADM

```

Figure 4-6 DB2 joblog messages

**Attention:** If you receive the IRR912I message during initialization, your exit is not active and native DB2 authorization will be used.

## 4.3 Protecting DB2 objects with RACF profiles

Resource access control can be managed through a DB2 built-in mechanism. Two SQL statements are used to provide security control: GRANT and REVOKE. GRANTS are recorded in the DB2 catalog.

Every process that connects to or signs onto DB2 is represented by one ID, called the primary authorization ID. All other IDs are secondary authorization IDs. One ID, either primary or secondary, is designated as the CURRENT SQLID.

DB2 controls access to its objects by a set of privileges. The GRANT statement grants privileges to authorization IDs. Privileges can be explicit or implicit.

Explicit privileges have names and are held as a result of GRANT and REVOKE statements. There is a set of privileges for each type of DB2 object. An example is shown in Figure 4-7.

```
GRANT SELECT ON TABLE SYSIBM.SYSDATABASE TO USER1
```

Figure 4-7 Example of a native DB2 GRANT to authorize access to a database

Implicit privileges are related to ownership of an object and include an ability to alter or drop an object or insert or delete a row.

### 4.3.1 DB2 object privileges

DB2 controls the access to its objects by a set of privileges. Sets of privileges are grouped into administrative authorities. These authorities form a hierarchy. Each hierarchy includes a specific group of privileges, and each privilege allows an action on an object. There is a set of privileges for each type of DB2 object. For example, the privileges for a table are:

<b>ALTER</b>	Change the table definition.
<b>DELETE</b>	Delete rows in a table.
<b>INDEX</b>	Create an index on a table.
<b>INSERT</b>	Insert rows into the table.
<b>REFERENCE</b>	Add or drop a referential constraint referring to the table.
<b>SELECT</b>	Retrieve data from the table.
<b>TRIGGER</b>	Define a trigger on the table.
<b>UPDATE</b>	Change the contents of a specific column.

When using RACF to manage security for DB2, RACF provides a member class and a grouping class for each DB2 object, for example:

<b>MDSNTB</b>	The member class for a table is MDSNTB. Member classes represent single objects or multiple similar named objects if generics are used. A member class starts with the character M.
<b>GDSNTB</b>	The grouping class for tables is GDSNTB. Grouping classes can represent one or more objects that have similar security requirements. Grouping classes start with the letter G.

A complete list of RACF member and grouping classes for DB2 is presented in Table 4-3.

*Table 4-3 Relationship between RACF profiles and DB2 objects*

RACF class	DB2 object	DB2 privilege	RACF profile
MDSNDB GDSNDB	Database	CREATE TAB	DB2-subsystem.database-name.CREATETAB
MDSNTB GDSNTB	Table	DELETE	DB2-subsystem.table-owner.table-name.DELETE
MDSNTB GDSNTB	Table	ALTER	DB2-subsystem.table-owner.table-name.ALTER

**Attention:** There is a DB2 to RACF migration tool, which was internally developed by IBM, but which is not officially supported. This can be found at:

<http://www.ibm.com/servers/eserver/zseries/zos/racf/racfdb2.html>

Table 4-4 RACF resources that are checked when a user issues a *SELECT* command

RACF profile name	Class	Note
DB2-subsystem.table-owner.table-name.SELECT	MDSNTB	Gives access to the table
DB2-subsystem.database-name.DBADM	DBADM	Gives access to the database that holds the table
DB2-subsystem.SYSCTRL	DBADM	Bypassed for user tables
DB2-subsystem.sysadm	DBADM	-

### 4.3.2 Mapping DB2 authorization checks

The RACF access control module maps the required DB2 authorization into RACF profiles. It builds resource names depending on the classification model being used, as follows:

**Single system scope**            *[objectname]privilegename*

**Multiple system scope**        DB2-subsystem-name.[objectname]privilege-name

#### Comparison of a CREATE table

In DB2, the DBMAINT, DBCTRL, and DBADM administrative authorities are sufficient for the CREATETMTAB privilege. However, with the RACF external security module, a user must have one of the following:

- ▶ The CREATETMTAB privilege
- ▶ The CREATETAB privilege
- ▶ SYSCTRL authority
- ▶ SYSADM authority

The user must have sufficient authority to:

- ▶ DB2-subsystem.database-name.CREATETAB in RACF class MDSNDB
- ▶ DB2-subsystem.database-name.DBMAINT in RACF class DSNADM
- ▶ DB2-subsystem.database-name.DBCTRL in RACF class DSNADM
- ▶ DB2-subsystem.database-name.DBADM in RACF class DSNADM
- ▶ DB2-subsystem.SYSCTRL in RACF class DSNADM

**Note:** If any one of the authorization checks is successful, access is allowed.

#### Long name support

DB2 has extended the lengths that may be specified for many of its constructs, which resulted in the following:

- ▶ The requirement for longer RACF resource names.
- ▶ Several RACF general resource classes have been updated to support these longer names; MDSNTB, DSNADM, MDSNCL, MDSNSG, MDSNUT, MDSNUF, MDSNSC, MDSNSP, and MDSNJR now have a maximum profile length of 246 characters.
- ▶ SCHEMA names are truncated at 100 characters when building a RACF resource name.

**Note:** An access level of READ is all that is required for RACF authorization to a DB2 resource profile.

**Grant on a Table**

```
PERMIT DB2-subsystem.SYSIBM.SYSDATABASE.ALTER ID(USER01) ACCESS(READ) CLASS(MDSNTB)
SETROPTS RACLIST(MDSNTB) REFRESH
```

Figure 4-8 Example of RACF commands to allow ALTER access on a table

**Note:** With DB2 Version 9, the RACF access control module provides the ability to use the RACF WARNING mode to assist in the implementation.

Figure 4-9 demonstrates how RACF groups can be used to protect DB2 objects that have the same access requirements.

```
RDEF GDSNTB DSN8810_TABGROUP UACC(NONE)
      ADDMEM(DB8L.DSN8810.ACT.DELECT, +
            DB8L.DSN8810.DEPT.SELECT)
RALT GDSNTB DSN8810_TABGROUP ADDMEM(DB8L.DSN8810.PROJ.SELECT)
PERMIT DSN8810_TABGROUP CLASS(GDSNTB) ID(USER01) ACC(READ)
SETROPTS RACLIST(GDSNTB) REFRESH
```

Figure 4-9 RACF commands to populate the GDSNTB class

Use the CONNECT command to add users who have the same access requirements to a common user group. In the example shown in Figure 4-10, a group is created called DB8LPROJ with the ADDGROUP command, and then use the CONNECT command to add USER01 to that group.

```
//STEPA EXEC PGM=IKJEFT01,DYNAMNBR=20
//SYSTSPRT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSTSIN DD *
      ADDGROUP DB8LPROJ OWNER(SYS1) SUPGROUP(SYS1)
      CONNECT USER01 GROUP(DB8LPROJ) AUTHORITY(USE)
/*
```

Figure 4-10 Sample RACF commands to create a user group and connect a user to it

## Protecting administrative authorities

As previously mentioned, sets of privileges are grouped into administrative authorities which form a hierarchy. Each hierarchy includes a specific set of privileges. The administrative authorities fall into the categories of system, database, and collection authorities. The highest ranking administrative authority is SYSADM. The administrative authorities are as follows:

- SYSADM** System administrator authority includes all DB2 privileges except a few that are reserved for install SYSADM.
- SYSCTRL** System controller authority includes all DB2 privileges except read or modify user data.
- SYSOPR** System operator includes all DB2 privileges except those that read or modify data.

An example of the use of the administrative authorities is shown in Figure 4-11 on page 45.



```

SYSOPR
RDEF DSNADM DB2-subsystem.SYSOPR UACC(NONE)
PERMIT DB2-subsystem.SYSOPR ID(USER01) ACCESS(READ) CLASS(DSNADM)
SYSADM
RDEF DSNADM DB2-subsystem.SYSADM UACC(NONE)
PERMIT DB2-subsystem.SYSADM ID(USER01) ACCESS(READ) CLASS(DSNADM)
SYSCTRL
RDEF DSNADM DB2-subsystem.SYSCTRL UACC(NONE)
PERMIT DB2-subsystem.SYSCTRL ID(USER01) ACCESS(READ) CLASS(DSNADM)

```

Figure 4-11 Defining profiles for administrative authorities

**Note:** All DB2 administrator authorities should be defined with a UACC(NONE) before you activate the RACF access control module. You can then selectively authorize users at a higher level by executing the PERMIT command.

The database authorities are as follows:

- DSNADM** Database administrative authority includes privileges required to control a database, including alter or drop table spaces, tables or indexes. Profiles that protect DB2 objects are shown in Table 4-5.
- DSNCTRL** Database administrative authority includes privileges required to control a database and run utilities against the database.
- DBMAINT** Database maintenance authority.
- PACKADM** Package administrator has all package privileges in specific collections.

Table 4-5 RACF profiles in the DSNADM class that protect system administrative authorities

DB2 object	DB2 privilege	RACF profile
SYSTEM	SYSADM SYSCTRL SYSOPR	DB2_subsystem.SYSADM DB2_subsystem.SYSCTRL DB2_subsystem.SYSOPR
DATABASE	DBADM DBCTRL DBMAINT	DB2_subsystem.database_name.DBADM DB2_subsystem.database_name.DBCTRL DB2_subsystem.database_name.DBMAINT
All collections	PACKADM	DB2_subsystem.PACKADM
Specific collections	PACKADM	DB2_subsystem.collection_id.PACKADM

**Restriction:** DB2 administrative authorities Install SYSADM and Install SYSOPR are defined in the DSNZPARM. The RACF access control module is not invoked for these user IDs and therefore it cannot be used to control access to these authorities.

### 4.3.3 Preventing cascading revoke

The RACF remove ID utility, IRRRID00, can help you keep the RACF database current. You can use this utility to remove all references to group names and user IDs that no longer exist in or are about to be removed from the RACF database. Also, you can specify a replacement ID for those IDs that will be removed.

Using native DB2 security when access is revoked from a certain DB2 authorization ID, all access granted by that authorization ID to the same resource is revoked as well. RACF offers a solution to this problem.

### **Example 1**

- ▶ USER01 GRANTS a privilege P to USER02 on an object.
- ▶ USER02 GRANTS the same privilege P to USER03 on the same object.

When USER01 REVOKES the privileges P from USER02, USER03 also loses its privilege P on that object.

### **Solution 1**

With the RACF access control module enabled the table is protected using a RACF profile. In this scenario USER01 would be in the RACF profile access list alongside USER02 and USER03. When User01 leaves, his user ID is removed from the access list as part of standard RACF cleanup using the utility IRRRID00.

### **Example 2**

USER01 creates a table called MYTABLE. Since USER01 created the table using his user ID and primary auth-id, he will own the object. He automatically gains all privileges on that table, and because he is the creator of the table he can GRANT privileges to USER02 and USER03.

USER01 leaves the company; his user ID is deleted from RACF and the DBA is requested to REVOKE all of USER01's privileges. The problem is the only way to REVOKE all the privileges that USER01 gained by creating the table is to drop and recreate the table. This involves saving the data. All the privileges granted to users and groups will need to be captured so that they can be regranted. In addition, all plans and packages depending on that table will have to be rebound. This is a risky business and even if it goes well it can lead to extended application outages.

### **Solution 2**

With the RACF access control module enabled the table is protected using a RACF profile. In the described scenarios USER01 would be in the RACF profile access list alongside USER02 and USER03.

Using the PERMIT command you can remove the users' access, in both examples, from the access lists in the RACF profile. Figure 4-12 on page 47 shows the RACF profile and access list before the PERMIT command was executed.

CLASS	NAME			
-----	----			
MDSNTB	** (G)			

GROUP	CLASS	NAME		
-----	-----	----		
GDSNTB				

LEVEL	OWNER	UNIVERSAL ACCESS	YOUR ACCESS	WARNING
-----	-----	-----	-----	-----
00	GGAINS	NONE	ALTER	NO
.				
.				
.				
USER	ACCESS			
----	-----			
GGAINS	ALTER			
USER01	READ			
USER02	READ			
USER03	READ			

Figure 4-12 Before USER01's access is revoked

By executing the JCL shown in Figure 4-13, USER01 is removed from the access list but the DB2 OBJECT is still protected by the profile and no other user has been affected by deleting USER01.

```
//STEPS EXEC PGM=IKJEFT01,DYNAMNBR=20
//SYSTSPRT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSTSIN DD *
PE ** ID(USER01) CLASS(MDSNTB) DELETE
SETROPTS RACLIST(MDSNTB) REFRESH
```

Figure 4-13 Removing a users access to a resource

**Note:** Do not forget to issue the SETROPTS command to activate any changes made to a RACF resource profile.

CLASS	NAME			
-----	----			
MDSNTB	** (G)			
GROUP	CLASS	NAME		
-----	-----	----		
GDSNTB				
LEVEL	OWNER	UNIVERSAL ACCESS	YOUR ACCESS	WARNING
-----	-----	-----	-----	-----
00	GGAINS	NONE	ALTER	NO
.				
.				
.				
USER	ACCESS			
----	-----			
GGAINS	ALTER			
USER02	READ			
USER03	READ			

Figure 4-14 After USER01's access has been revoked

### Deleting user IDs

When you remove USER01 from the access list of a specific profile, the user ID will still exist. Should USER01 no longer require his user ID, it should be deleted and removed from the access list as part of RACF cleanup using the RACF utility IRRRID00.

The remove ID utility processes the output of the RACF database unload utility (IRRDBU00). You can use the remove ID utility to:

- Find all residual references to user IDs and group names that no longer exist in the RACF database.
- Find all references to a list of user IDs and group names that you specify in the SYSIN file.

For additional information, see *z/OS Security Server RACF Systems Administrator's Guide*, SA22-7683.

**Note:** The RACF IRRRID00 utility should be used to DELETE users and any of their associated RACF profiles from the RACF database.

## 4.4 Authorization checking

The RACF access control module allows you to use RACF as an alternative to DB2 authorization checking for DB2 objects, authorities, and utilities. When DB2 has an authorization request, DB2 calls the DSNX@XAC exit point using a parameter list, defined by DSNDXAPL.

DB2 provides an exit point that lets you provide your own access control authorization exit routine, or lets RACF or an equivalent security system perform DB2 authorization checking. Your routine specifies whether the authorization checking should all be done by RACF only, or by both RACF and DB2. (Also, the routine can be called and still let all checking be

performed by DB2.) When DB2 invokes the routine, it passes three possible functions to the routine:

- ▶ Initialization (DB2 startup)
- ▶ Authorization check
- ▶ Termination (DB2 shutdown)

The bulk of the work in the routine is for authorization checking. When DB2 must determine the authorization for a privilege, it invokes your routine. The routine determines the authorization for the privilege and then indicates to DB2 whether the privilege is authorized or not authorized, or whether DB2 should do its own authorization check instead.

### Exit routine bypassed

In the following situations, the exit routine is not called to check authorization:

- ▶ The user has installation SYSADM or installation SYSOPR authority (where installation SYSOPR authority is sufficient to authorize the request). This authorization check is made strictly within DB2.
- ▶ DB2 security has been disabled. (You can disable DB2 security by specifying NO on the USE PROTECTION field of installation panel DSNTIPP).
- ▶ Authorization has been cached from a prior check.
- ▶ In a prior invocation of the exit routine, the routine indicated that it should not be called again.
- ▶ When GRANT statements are used.

**Note:** For more information about how to use the routine that is provided, see *DB2 Universal Database for z/OS RACF Access Control Module Guide*, SC18-7433.

The DSNDXAPL provides information for the exit, such as:

- ▶ The privilege that is being requested
- ▶ The object type of the privilege
- ▶ Owner or schema name of the object
- ▶ Object information to help determine security

Based on the privilege and object type, a mapping can be found to determine what kind of checks need to be done.

- ▶ Ownership or match checks, or both (if required)
  - Basic string compare is done, so there is no call to RACF
- ▶ Object authorities checks (if required)
  - Profile is built using the object information in DSNDXAPL
  - One or more FASTAUTH calls are made to RACF
- ▶ Administrative authorities checks
  - A profile is built using the object information in DSNDXAPL
  - A FASTAUTH call is then made to RACF

The RACF external security module requires an input ACEE to perform authority checking. When an input ACEE, XAPLACEE, is not provided to the RACF access control module, it produces a return code of 4, which indicates that DB2 should defer to DB2 security for that particular authorization check. For these requests, authority checking must be implemented

using the DB2 GRANT and REVOKE commands. Any RACF profiles defined for these requests will not be used. RACF access control module processing ends when the FASTAUTH returns a return code of 0 or it has exhausted all of its checks.

For a detailed description and a list of what resources for authorization are checked, refer to Appendix C and Appendix D of the *DB2 UDB for z/OS V8 RACF Access Control Module Guide*, SC18-7433.

**Note:** Only READ authority is needed for the check to pass.

## RACF access control module return codes

The RACF access control module will not generate a failure until after it has checked the entire list of profiles. The following return codes are possible:

► Return Code 0 - Access permitted

Upon the first successful check, the RACF access control module will pass back a return code 0 and access is allowed.

- This will override all return codes that follow.
- This may produce an SMF record, depending on the setting on the profile that covers the resource.

Reason code meaning:

- |    |   |
|----|---|
| 0  | Access permitted by FASTAUTH checking.  |
| 13 | Access permitted by implicit privilege of ownership.  |
| 14 | Access permitted because current SQL ID matches schema name.  |
| 16 | Access permitted because the role associated with the request owns the object.                      |
| 17 | Access permitted because the authorization ID associated with the request owns the implicit object. |
| 18 | Access permitted because the role associated with the request owns the object.                      |

► Return code 4 - Unable to determine; perform DB2 authorization checking

If no object checks are done, and one of the administrator checks results in a return code 4, the RACF Access Control Module passes back a return code 4.

- No RACF profile is found for the resource.
- If all the object checks result in a return code 4, the RACF Access Control Module passes back a return code of 4 and authorization is deferred to DB2.
- This does not produce an SMF record.

Reason code meaning:

- |    |  |
|----|--|
| 0  | Input class (XAPLTYPE) not active.   |
| 11 | Input ACEE (XAPLACEE) not provided.  |
| 14 | The ALET could not be created for cross memory ACEE.   |
| 15 | Input privilege code (XAPLPRIV) or input class (XAPLTYPE) not defined to the RACF access control module.                     |
| 16 | Input privilege code (XAPLPRIV) does not contain any rules.  |
| 18 | Issued when running on z/OS V1R7 and trying to create an object in a trusted context with the "role as object owner" clause. |

► Return Code 8 - Access denied

If at least one object check results in a return code 8, the RACF Access Control Module passes back a return code 8 and access is denied.

If no object checks are done, and all the administrator checks result in a return code 8, the RACF access control modules pass back a return code 8.

- RACF produces ICH408I messages for the first profile in the list of profiles.
- An SMF record is written if requested on the covering profile.

Reason code meaning:

- 0 Access denied.
- 17 Autobind indicator (XAPLAUTO) is not zero, indicating AUTOBIND was requested. Manual REBIND is required.
- 100 Role information was passed, but ignored because the RACF access control module was assembled with z/OS V1R7 macros.

## 4.5 Auditing considerations

The RACF profiles which represent the various DB2 privileges also enable you to use the RACF auditing tools to extract information.

One of the first things that a SAF/RACF-secured application needs to do is establish the user's security credentials, such as build an ACEE, which traditionally is done via invocation of RACROUTE REQUEST=VERIFY.

An invoker of RACROUTE REQUEST=VERIFY can also supply the audit information itself on a CREATE request. This ability could be very important to an application that has access to the initial authentication information from a different user registry and wants to ensure that the information will be audited.

The new audit information consists of the parameters shown in Table 4-6. This is what will be included in the SMF records written by RACROUTE REQUEST=AUTH and RACROUTE REQUEST=FASTAUTH when the ACEE has been extended with an ICTX block.

*Table 4-6 New audit information*

Parameter	Value
Authenticated user name	Maximum length of 510 bytes.
Registry name	Maximum length of 255 bytes.
Host name	Maximum length of 128 bytes.
Authentication mechanism	An authentication mechanism object identifier (OID). The authentication mechanism has a maximum length of 16 bytes.

RACROUTE REQUEST=AUTH and RACROUTE REQUEST=FASTAUTH now include the ICTX information in SMF audit records when available.

RACROUTE REQUEST=EXTRACT,TYPE=ENVRXTR is extended to ICTX information in the ENVR object, when available.

### RACF and SMF records

RACF will not produce an SMF record for failures if:

- The requester meets any of the requirements and access is granted
- The RACF access control module returns authority checking to DB2
- An audit record is produced for the first resource that has auditing indicated and receives a return code of 8

RACF produces an SMF record for successful access when the RACF profile indicates that should be performed. When RACF builds an SMF record for any audit event, not just job initiation, if the ACEE points to an ICTX block, new extended relocate sections will be built for the new information, as follows:

```
RACROUTE REQUEST=VERIFY,ENVIR=CREATE, ...ICTX=ictx-block addr
```

**Note:** If DB2 objects were defined to RACF using the WARNING option, you will see ICH408I messages that identify those profiles that would fail a request, and the request would be allowed.

You can use the SMF data unload utility or the RACF report writer to extract and format the SMF records.

**Note:** The SETROPTS LOGOPTIONS(classname) cannot be used with the RACF/DB2 support.

### 4.5.1 Debugging considerations

If access is denied by RACF, an ICH408I message is produced. Of more concern is when access is unexpectedly allowed to a DB2 object. It is possible to diagnose the cause of this by examining the log stream data that is produced as part of the RACROUTE REQUEST=FASTAUTH processing. The log stream data contains additional diagnosis information and that can be linked to the DB2 trace record IFCID 314 and the corresponding RACF SMF record.

The log stream data consists of information that can help you audit DB2 successfully. DB2 uses the XAPL parameter list (DSNDXAPL macro) to pass log stream information to the RACF access control module.

RACROUTE REQUEST=FASTAUTH Logstr=parameter contains the input portion of XAPL. It does the following:

- ▶ Identifies the RACF access control module request that caused RACF to create the audit record
- ▶ Links SMF type 80 records to DB2 IFCID 314 records

By examining these records it is possible to identify why the request was authorized.

For a more in-depth discussion refer to the *DB2 UDB for z/OS V8 RACF Access Control Module Guide*, SC18-7433.

## 4.6 Multilevel security

Multilevel security is a security policy that allows for the classification of data and users based on a system of hierarchical security levels combined with a system of non-hierarchical security categories. It aims to keep information classified and compartmentalized between users. It uses a combination of DAC and MAC controls.

### Discretionary access controls (DAC)

DAC is access granted at the discretion of the owner through the use of access lists.



## **Mandatory access control (MAC)**

Data is accessed based on a comparison of the classification of the user and the classification of the data. MAC prevents users from accessing information as follows:

- ▶ A classification they are not authorized to.
- ▶ Changing the classification of information they do have access to by writing down.
- ▶ The system does not allow a user to view rows of data in DB2 that he is not classified to view.
- ▶ MAC is based on the theory of dominance and is implemented through the use of security labels.

RACF has several options that can be turned on and off to manipulate different aspects of a multilevel security environment. It is possible to have a partial multilevel security system which would take advantage of just some of the available features.

There are several major principles we need to consider. They are:

- ▶ Security labels and the relationships between security labels, specifically:
  - Dominance
  - Equivalence
  - Disjoint
- ▶ The type of access requested (MAC Access):
  - Read only
  - Read write
  - Write only

### **Read-only test**

The user intends to only read information; therefore, he should be able to read information at his own classification or information at a lower classification. In terms of MAC checking, the user's security label must dominate the security label of the object that the user intends to read. This allows the user to read information covered by his security label, but not information of a higher level, or outside his category.

### **Write-only test**

The user is intending to only write information; therefore, he should be able to write information at his classification or information at a higher classification. In terms of MAC checking, the security label of the object that he intends to write to must dominate his security label. This allows him to write information at his security label or higher, but not information of a lower level, or outside his category (that is, he will not be able to declassify information).

### **Read/write test**

The user intends to read and write information. Like in the read-only case, the user's security label must be able to dominate the object's security label; and, like the write-only case, the object's security label must be able to dominate the user's security label. Therefore, for the user to do a read/write action to the object, the user's and object's security labels must both be equivalent to each other. This allows the user to read and write information only at his security label, but not outside his security label (that is, he will not be able to declassify information).

## Controlled write-down

There might be cases where you want to allow for controlled situations of write-down. The security administrator can assign a write-down by user privilege to individual users or groups of users that allows them to select the ability to write down. The security administrator activates and deactivates the privilege by creating a profile in the FACILITY class called IRR.WRITEDOWN.BYUSER. A user can activate write-down mode if the profile exists, and the user has at least read access to it. If the user has update or higher access to the profile, write-down mode is active by default when the user enters the system.

**Note:** RACF provides the RACPRIV command and z/OS UNIX provides the **writedown** command, which allow users who are authorized to the write-down privilege to reset and query the setting of their write-down mode.

## MAC principles

When SETROPTS MLS is active in your environment, users are limited in their WRITE actions, such as their authority to copy data from a resource with one security label to a resource with a lower security label. If you need to allow certain users to have this authority, also called the writedown privilege, you can authorize them using a FACILITY class profile called IRR.WRITEDOWN.BYUSER.

To summarize the basic MAC principle with MLS active, as illustrated in Figure 4-15:

- ▶ If the user's security label dominates the data's security label, the user can read the data.
- ▶ If the data's security label dominates the user's security label, the user can write the data.
- ▶ If user's and data's security labels are equivalent, then the user can read and write to the data.
- ▶ If there are no equivalence or dominance relationship between the user's security label and data's security label, the user will be denied access to the data.

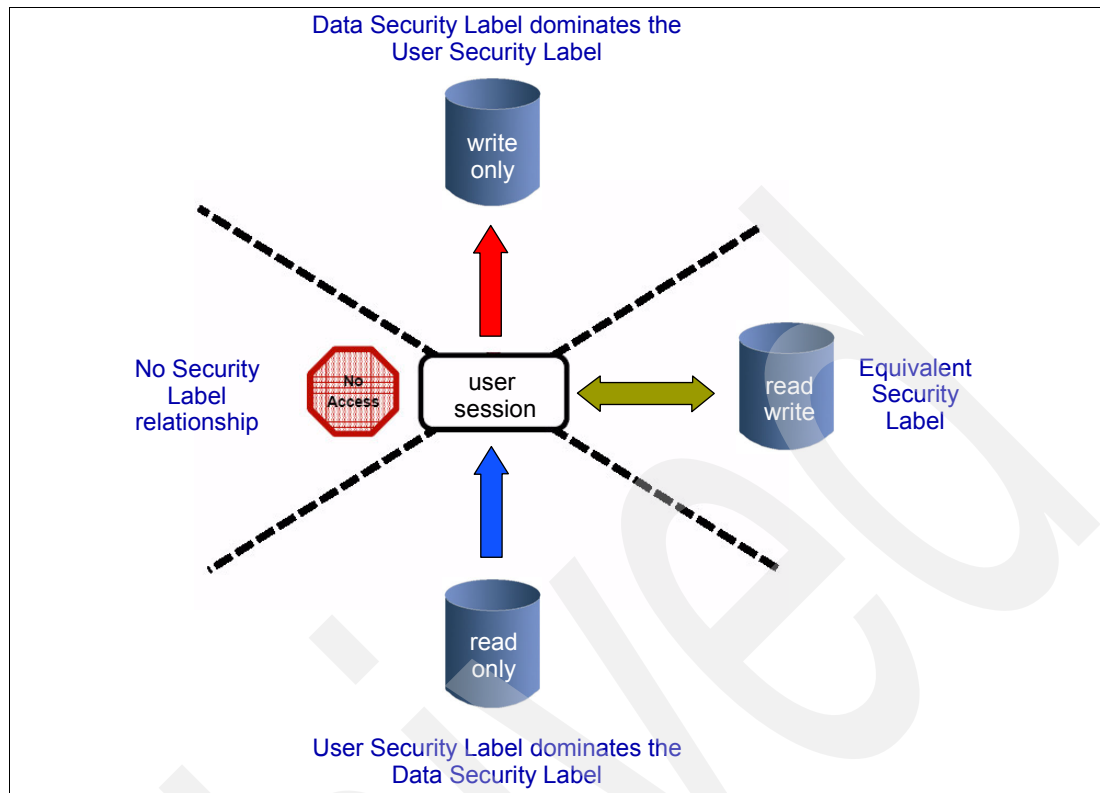


Figure 4-15 MAC principles

### Defining security labels

The security administrator uses RACF to control each subject's access to information by specifying which security label the subject can use.

A security label establishes an association between a RACF security level and a set of zero or more RACF security categories, Group A, Group B, and Group C.

You must define two profiles in the RACF SECDATA resource class: one to define the security levels and the other to define the security categories for the system.

### 4.6.1 DB2 and multilevel security

Multi level security can be implemented in two ways in DB2:

- ▶ At the object level
- ▶ At row level

The relationship between DB2 users and DB2 objects is important to understand and is summarized as follows:

- ▶ A DB2 user is an entity that requires access to system resources, and can be a:
  - Human user
  - Stored procedure
  - Batch job
- ▶ An object is any system resource to which access must be controlled, such as a:
  - Data set
  - Table
  - Row

- Command

Using multilevel security you can define security for DB2 objects by assigning security labels in the RACF SECLABEL class to your objects. You can define a hierarchy between those objects. Multilevel security restricts access to any object based on the security label of that object.

#### 4.6.2 Multilevel security with row-level granularity

DB2 also has an implementation of multilevel security at row level. This allows us to perform row-level security checks. The traditional DB2 response to this requirement has been to use views. However, views have several disadvantages. RACF offers you the ability to define profiles in the SECLABEL class. The security level and category used in SECLABEL must be defined with a SECLEVEL and CATEGORY profile in the RACF SECDATA class.

For more information on Multilevel security and its RACF implementation refer to the following publications:

*z/OS Planning for Multilevel security and Common Criteria*, GA22-7509

*Multilevel Security and DB2 Row-Level Security Revealed*, SG24-6480



## RACF virtual key ring support

This chapter describes the virtual key ring function that has been added to z/OS V1R8 to support enhancements for key rings.

The following topics are included:

- ▶ RACF and key rings
- ▶ Virtual key rings
- ▶ Real key rings

## 5.1 RACF and key rings

A key ring is a collection of certificates that identify a networking trust relationship. In a client-server network environment, entities identify themselves using digital certificates. Server applications on z/OS that wish to establish network connections to other entities can use RACF key rings and other related services to determine the trustworthiness of the client or peer entity.

Key rings contain the public keys that are associated with signers of certificates. These public keys are, in reality, contained in certificates themselves. Therefore, verifying one certificate requires the use of a different certificate, the signer's certificate. In this fashion, a chain of certificates is established, with one certificate being verified by using another certificate and that certificate being verified by yet another certificate, and so on. A certificate, and its associated public key, can be defined as a *root* certificate. A root certificate is self-signed, meaning that the public-key contained in the certificate is used to sign the certificate. Using a root certificate implies that the user trusts the root certificate.

Key rings are associated with specific RACF user IDs and a RACF user ID can have more than one key ring. Key rings are managed using the RACDCERT RACF command, and are maintained in the general resource class DIGTRING.

### RACDCERT command

The RACDCERT command is your primary administrative tool for managing digital certificates using RACF. Authority to use the RACDCERT command is controlled through resources in the FACILITY class. The RACDCERT command is used to manage resources in the following classes:

<b>DIGTRING</b>	Profiles in the DIGTRING class contain information about key rings and the certificates that are part of each key ring. Key rings are named collections of the personal site and certificate authority certificates associated with a specific user.
<b>DIGTNMAP</b>	Profiles in the DIGTNMAP class contain information about certificate name filters.
<b>USER</b>	Profiles in the USER class contain information about digital certificates that are associated with the user.

### 5.1.1 Secure Sockets Layer

Secure Sockets Layer (SSL) and other security middleware use the R\_datalib callable service (IRRSDL00), which provides the function required to implement the Open Cryptographic Services Facility Data library functions, to retrieve certificate information from RACF. These certificates must be connected to a RACF key ring in order for applications to retrieve certificates and private keys from RACF.

SSL is a base component of z/OS. It is a generic set of API applications that can be used to protect TCP/IP socket communications using the Secure Socket Layer (SSL), and also the Transaction Layer Security (TLS) protocol.

The key ring is the data store that R\_datalib callable service opens, reads, and closes as directed by the application. With z/OS V1R8, these applications can also read virtual key rings using R\_datalib.

### 5.1.2 R\_datalib (IRRSDL00) callable services

The R\_datalib service provides the function required to implement the Open Cryptographic Services Facility Data library functions. Authorized applications, such as servers, which invoke the R\_datalib callable service (IRRSDL00), can extract private keys and manage certificate serial numbers. You authorize these applications for these functions by administering the same FACILITY class resources checked by the RACDCERT command.

### 5.1.3 What is a virtual key ring

A virtual key ring is the set of certificates used by a user or server application to determine the trustworthiness of a client or peer. Real key rings must be created and populated as required by the application. However, a virtual key ring does not need to be added to RACF. Each RACF user ID is associated with a virtual key ring. The most common type is the CERTAUTH virtual key ring, which contains all the trusted CA certificates defined on the system. It is used when an application validates the certificates of others but has no need for its own certificate and private key.

### 5.1.4 Problems before z/OS V1R8

Customers running on pre-V1R8 z/OS needed to create individual key rings for each application or user ID, usually containing the same set of certificate authorities (CA), and therefore the RACF administrator had to create and populate a key ring for each user, and in most cases the key rings were identical.

Secure Sockets Layer (SSL) client applications still need a key ring to identify trusted certificate authorities; those applications which do not perform client authentication, do not need their own certificate.

### 5.1.5 RACF virtual key ring benefits

Virtual key ring allows all certificates under CERTAUTH to be used as a “virtual” key ring. In this way, applications can use this key ring instead of their own key ring, eliminating unnecessary administration.

### 5.1.6 How to use a virtual key ring

For applications using SSL, such as z/OS FTP, or other middleware programs that read RACF key rings through the R\_datalib callable service, a virtual key ring can be specified in place of a real key ring, whenever a real key ring is expected. Therefore, virtual key rings can be accessed through R\_datalib callable service like real key rings.

To include virtual key rings, the application user specifies an asterisk (\*) for the key ring name along with the owning user ID using the form `<owning-id>/*`.

Specifying “\*” for the Ring\_name means a virtual key ring. The virtual key ring can be qualified with the user ID (`<user-id>/<ring-name>`), then `*AUTH/*` would be the CERTAUTH virtual key ring.

Thus, when using a virtual key ring, replace the default value with either `*AUTH/*` or `*SITE/*` to point to all your trusted CA or site certificates, respectively.

**Remember:** The name of a virtual key ring is always an asterisk.

### Example

This example shows the use of a virtual key ring with a z/OS FTP client with Transaction Layer Security (TLS).

A z/OS FTP user can use the FTP server's virtual CERTAUTH key ring for authentication when all of the following conditions are true:

- ▶ The user has a KEYRING directive in his FTP.DATA file specified as follows:  
KEYRING \*AUTH\*/\*
- ▶ The user directs FTP to use TLS by specifying **-a TLS** or **-r TLS** on the FTP command:  
ftp -r TLS ftp.ibm.com
- ▶ Client authentication is not required and the virtual key ring is used only to authenticate the FTP server.

## 5.1.7 Virtual key ring usage and invocation

The R\_datalib callable service determines who has authority over a virtual key ring, as well as to a real key ring, by checking the resource IRR.DIGTCERT.LISTRING in the FACILITY class.

- ▶ **READ** authority in this resource provides a user with the ability to read his own virtual key ring.
- ▶ **UPDATE** authority in this resource provides a user with the ability to read his own virtual key ring as well as other user's virtual key ring.

**Note:** CERTAUTH and SITE virtual key rings are considered owned by everyone; therefore, only READ authority is required. This means that any user that is using a real key ring can make use of the CERTAUTH virtual key ring without needing to make any RACF authority change.

Since virtual key rings are not real key rings, you cannot RACDCERT key ring functions to manage them. For example, you cannot issue the command RACDCERT CERTAUTH LISTRING(\*). However, you can issue the RACDCERT CERTAUTH LIST.

## 5.1.8 Virtual key ring implementation

Consider the following when implementing a virtual key ring:

- ▶ Deal with all certificates that are installed under a given user ID as a virtual key ring owned by that user ID.
- ▶ Allow an indicator (\*) that a virtual key ring is requested.
- ▶ Allow special user IDs CERTAUTH (irrcerta) and SITE (irrsitec) to own virtual key rings.
- ▶ Implement access control, as follows:
  - Allow access to virtual key rings by their owners following rules in place for regular key rings for resource IRR.DIGTCERT.LISTRING in the FACILITY class.
  - Key rings owned by CERTAUTH or SITE user IDs are accessible by any user ID that can access their own key ring.
  - READ access allows users to read their own virtual key ring.
  - UPDATE access allow users to read other users' virtual key rings.



## 5.1.9 Real key ring and virtual key ring

Although to a System SSL a virtual key ring is similar in appearance to a real key ring, they do have some differences, which are described in Table 6-2.

Table 5-1 Differences between real and virtual key rings

Real key ring	Virtual key ring
A certificate's ring usage is set when the certificate is connected to the key ring.	All certificates within the ring have the same usage as follows: <ul style="list-style-type: none"><li>▶ CERTAUTH for the CERTAUTH virtual key ring (RACF reserved user ID irrcerta or *AUTH*)</li><li>▶ SITE for the SITE virtual key ring (RACF reserved user ID irrsitec or *AUTH*)</li><li>▶ PERSONAL for the virtual key rings of all other non-reserved user IDs</li></ul>
A private key is only returned when the certificate's ring usage is PERSONAL, and the caller's user ID is the user ID associated with the certificate profile or, for CERTAUTH and SITE certificates, the caller is RACF SPECIAL or has CONTROL authority to the resource IRR.DIGTCERT.GENCERT in the FACILITY class.	A private key is never returned from the CERTAUTH or SITE virtual key rings. However, a user can retrieve the private keys associated with the certificates in his or her own virtual key ring.
Resource IRR.DIGTCERT.LISTRING in the FACILITY class: <ul style="list-style-type: none"><li>▶ READ access allows a user to read its own key ring.</li><li>▶ UPDATE access allows a user to read its own and other users' key rings.</li></ul>	Resource IRR.DIGTCERT.LISTRING in the FACILITY class: <ul style="list-style-type: none"><li>▶ READ access allows a user to read its own virtual key ring.</li><li>▶ UPDATE access allows a user to read its own and other users' virtual key rings.</li></ul>

## 5.2 Related publications

The following publications reflect the virtual key ring new feature:

- ▶ *Security Server RACF Callable Services*, SA22-7691
- ▶ *Security Server RACF Security Administrator's Guide*, SA22-7683
- ▶ *Security Server RACF Command Language Reference*, SA22-7687
- ▶ *Cryptographic Services PKI Services Guide and Reference*, SA22-7693

Archived

## PKI Services

This chapter provides a brief overview of PKI Services and multiple CAs, and describes the function that has been added to z/OS V1R8 to support enhancements for z/OS Cryptographic Services PKI Services.

The following topics are included:

- ▶ PKI Services
- ▶ PKI Services multiple CAs overview
- ▶ Loosely-coupled CA examples
- ▶ Setup for PKI Services
- ▶ PKI Services support for SCEP
- ▶ PKI Services usage considerations

## 6.1 Introduction to PKI

The public key infrastructure (PKI) provides applications with a framework for performing the following types of security-related activities:

- ▶ Authenticate all parties that engage in electronic transactions.
- ▶ Authorize access to sensitive systems and repositories.
- ▶ Verify the author of any message through its digital signature.
- ▶ Encrypt the content of all communications.

The PKIX standard evolved from PKI to support the interoperability of applications that engage in e-business. Its main advantage is that it enables organizations to conduct secure electronic transactions without regard for the operating platform or application software package.

## 6.2 Overview of PKI Services

z/OS Cryptographic Services PKI Services allows you to use z/OS to establish a PKI infrastructure that serves as a certificate authority for your internal and external users, issuing and administering digital certificates in accordance with your own organization's policies.

PKI Services supports the following:

- ▶ Public Key Infrastructure for X.509 version 3 (PKIX).
- ▶ Common Data Security Architecture (CDSA).
- ▶ Delivery of certificates through the Secure Sockets Layer (SSL) for use with applications that are accessed from a Web browser or Web server.
- ▶ Delivery of certificates that support the Internet Protocol Security standard (IPSEC) for use with secure VPN applications or IPSEC.
- ▶ Delivery of certificates that support Secure Multipurpose Internal Mail Extensions (S/MIME), for use with secure e-mail applications.

### 6.2.1 Basic components of PKI Services and related products

The basic components of PKI Services and related products are shown in Table 6-1.

Table 6-1 Basic components of PKI Services and related products

Component	Purpose
Administration Web application	Assists authorized users to: <ul style="list-style-type: none"><li>▶ Review requests for certificates</li><li>▶ Approve or reject requests</li><li>▶ Renew or revoke certificates</li><li>▶ Review pending certificate requests</li><li>▶ Query pending requests to process those that meet certain criteria</li><li>▶ Display detailed information about a certificate or request</li><li>▶ Monitor certificate information</li><li>▶ Annotate the reason for an administrative action</li></ul>

Component	Purpose
End-user Web application	Guides your users to request, obtain, and renew certificates through their Web browsers.
Exit	Provides advanced customization for additional authorization checking, validating and changing parameters on calls to the R_PKIServ callable service (IRRSPX00) and capturing certificates for further processing.
ICSF (optional)	Securely stores the PKI Services certificate authority's private signing key.
LDAP	The directory that maintains information about the valid and revoked certificates that PKI Services issues in an LDAP.
PKI Services daemon	The server daemon that acts as your certificate authority, confirming the identities of users and servers, verifying that they are entitled to certificates with the requested attributes, and approving and rejecting requests to issue and renew certificates.
R_PKIServ callable service (IRRSPX00)	The application programming interface (API) that allows authorized applications, such as servers, to programmatically request the functions of PKI Services to generate, retrieve, and administer certificates.
RACF (or equivalent security software)	Controls who can use the functions of the R_PKIServ callable service and protects the components of your PKI Services system. RACF creates your certificate authority's certificate, key ring and private key.
z/OS HTTP Server	PKI Services use the Web server to encrypt messages, authenticate requests, and transfer certificates to intended recipients.

## 6.3 PKI Services multiple CAs overview

A certificate is used to prove identity. A secure server must have a certificate and a public-private key pair. A certificate is issued and signed by a certificate authority (CA). The default setup for PKI Services establishes the PKI Services certificate authority as a root CA, also known as a self-signed CA. Because there is no established trust hierarchy leading to a self-signed certificate, it is impossible to verify that a self-signed certificate is genuine. Accordingly, any person or application that wishes to process certificates issued by a root authority must explicitly trust the authenticity of the self-signed CA certificate.

Prior to z/OS V1R8, the PKI Services started task cannot be configured to have more than one signing CA certificate and key. On a single image, only one copy of the PKI Services started task can run at any given time. What this means is that users cannot operate more than one certificate authority per z/OS image. This prevents the following:

- ▶ Operating a certificate authority hierarchy
- ▶ Hosting multiple certificate authorities as a services provider

### 6.3.1 z/OS V1R8 enhancements

With z/OS V1R8, the restriction is removed that prevents multiple copies of the PKI Services started task, so now:

- ▶ Each started task instance can operate as a different CA.
- ▶ This allows customers to operate multiple CAs.

#### CA domain name

The CA domain name is used to qualify resources used by the particular CA as follows:

- ▶ Web page URLs (mixed case)
  - `http://webserver-name/Employees/public-cgi/camain.rexx`
- ▶ Data set qualifiers (uppercase and truncated to 8 characters)
  - VSAM ICL data set - `PKISERD.EMPLOYEE.VSAM.ICL`
- ▶ HFS pathnames (lowercase)
  - Configuration file - `/etc/pkiserv/employees/pkiserv.conf`

By adding a new CA domain, users have a unique URL and set of certificate templates to choose from, like an application domain, but they also have the services of their own CA, including the CA's certificate, signing key, VSAM data sets, and LDAP repository. Enabling multiple CAs is a natural extension for multiple application domains. Each CA domain can represent one instance of a CA, backed by a unique instance of the PKI Services daemon (and all its associated componentry), yet requiring no more than a single HTTP Server and a single set of CGIs.

#### Operation scenarios

There are two types of CA environments:

- ▶ Completely independent CAs  
Each CA has its own administrators, and end-users and administrators may not be aware of other CAs. An example of this is a services corporation that hosts CAs for other companies.
- ▶ Loosely coupled CAs  
There is one set of administrators for multiple CAs, and end-users may not be aware of other CAs, but administrators are aware. An example of this is a CA hierarchy.

### 6.3.2 Loosely coupled CA examples

Multiple PKIServ daemons may be started, each pointing to their own set of resources. Each set of resources is a CA domain (as opposed to an application domain). Loosely coupled mode is similar to multiple application domains except that each domain really is a separate CA instead of just looking like one, as shown in Figure 6-1 on page 67.

- ▶ Each CA has one unique certificates template file.
- ▶ Each end-user home page URL drives down to a separate PKIServ daemon courtesy of environment variables in the webserver address space that locate the certificates template file for the CA domain.

What makes it loosely-coupled is that there is only one set of PKI administrators for all the CA domains. A separate certificates template file is used by the administrators, which enables an administrator to select the CA to manage.

The environment variables for Figure 6-1 are as follows:

The diagram illustrates the architecture of Loosely-Coupled CAs. At the top, an **Admin** box is connected to a **Users** box. The **Users** box is connected to the **RACF** box via the URL `http://<server>/Customers/*`. The **RACF** box is connected to the **PKIServ Daemon - Customers** box via the URL `http://<server>/PKIServ/*`. The **RACF** box is also connected to the **PKIServ Daemon - Employees** box via the URL `http://<server>/Employees/*`. The **RACF** box is connected to the **RACF DB** box. The **RACF** box is connected to the **VSAM** and **LDAP** boxes. The **PKIServ Daemon - Customers** box is connected to the **VSAM** and **LDAP** boxes. The **PKIServ Daemon - Employees** box is connected to the **VSAM** and **LDAP** boxes. The **RACF** box is connected to the **VSAM** and **LDAP** boxes. The **RACF** box is connected to the **VSAM** and **LDAP** boxes. The **RACF** box is connected to the **VSAM** and **LDAP** boxes.

**Loosely-Coupled CAs**

**Admin**

**Users**

**Users**

**RACF**

**PKIServ Daemon - Customers**

**PKIServ Daemon - Employees**

**RACF DB**

**VSAM**

**LDAP**

**VSAM**

**LDAP**

`http://<server>/PKIServ/*`

`http://<server>/Customers/*`

`http://<server>/Employees/*`

`CADomain=CUSTOMER`

`CADomain=EMPLOYEE`

`CADomain=<as selected>`

`<APPLICATION NAME=PKISERV>`  
...  
`</APPLICATION>`

`<APPLICATION NAME=CUSTOMERS>`  
...  
`</APPLICATION>`

`/etc/pkiserv/employees/pkiserv.tmpl`

`<APPLICATION NAME=EMPLOYEES>`  
...  
`</APPLICATION>`

**S PKISERVD.CUSTOMER,DIR='/etc/pkiserv/customers'**  
`/etc/pkiserv/customers/pkiserv.conf`

**S PKISERVD.EMPLOYEE,DIR='/etc/pkiserv/employees'**  
`/etc/pkiserv/employees/pkiserv.conf`

*Figure 6-1 Loosely-coupled CAs*

Each CA instance is called a *CA domain*. They are similar to application domains that are already supported. When you want to operate more than one certificate authority (CA) on a single z/OS image, you must create a separate CA domain for each CA. Each CA domain uses its own daemon and operates as its own instance of PKI Services.

When you add CA domains, you can create a PKI infrastructure that contains subsets of end user populations (application domains), each supported by its own unique PKI Services application (PKI Services daemon and URL) and optionally by its own dedicated set of PKI administrators, as illustrated in Figure 6-1. If you already use multiple application domains, the key advantage of adding multiple CA domains is that you can build a certificate hierarchy of CAs and optionally provide certificate services to multiple organizations.

For each CA you add, you create a dedicated copy of the VSAM files, CA certificate, key ring, and LDAP namespace. You also create a dedicated copy of the PKI Services configuration file (pkiserv.conf), templates file (pkiserv.tmpl), and environment variables file

(pkiserv.envars), each in its own directory. You update the following CA-specific information in these files:

<b>pkiserv.conf</b>	Contains the CA-specific key ring name, VSAM data set names, and optionally CRLDistDirPath.
<b>pkiserv.envars</b>	Contains a variable <code>_PKISERV_CA_DOMAIN</code> to specify CA domain and the variable <code>_PKISERV_CONFIG_PATH</code> sets the directory for each CA domain.
<b>pkiserv.tmpl</b>	Contains the name of the end-user application section (default is CUSTOMER) that you rename to a CA-specific name, such as <code>&lt;APPLICATION NAME=CUSTOMERS&gt;</code> . It also contains the name of the administrative application section (default is PKISERV) that you can rename to a CA-specific name, such as <code>&lt;APPLICATION NAME=EMPLOYEES&gt;</code> .

To reconfigure an existing CA to tolerate other CAs, make the following changes:

- ▶ Update PKI and Web server environment variables.
- ▶ Define new FACILITY class protection profiles for R\_PKISERV, such as:  
    IRR.RPKISERV.PKIADMIN.ca-domain
- ▶ For each new CA, perform the previous procedures, plus the following:
  - Customize and run IKYSETUP REXX™ EXEC.
  - Copy initial CA pkiserv.envars, pkiserv.conf, and pkiserv.tmpl to a user directory and customize as needed:
    - Create VSAM data set names and HFS pathnames.
    - Add APPLICATION names in template file.
    - If loosely-coupled, add link and drop-down to the administrator page.
  - Update HTTP configuration files with new PROTECTION directives.
  - Allocate VSAM data sets.

**Note:** If you are already operating a PKI Services CA and you want to have multiple CA domains, you will first need to reconfigure the existing CA to tolerate the starting of additional CAs, as follows:

- ▶ Qualifying the IRR.RPKISERV.PKIADMIN resource with the CA domain enables the separation of administrators in independent mode.
- ▶ Qualifying the IRR.RPKISERV.\* resource with the CA domain enables the separation of end-users in independent and loosely coupled modes.
- ▶ The IKYSETUP REXX exec will create a lot of the resources needed by each new CA (for example, the CA's signing key and certificate). The existing CA will continue to use its existing resources.
- ▶ The PROTECTION directives in the HTTP configuration files enable the separate domain names in the URLs. The procedure is the same as that for adding a new application domain.

## 6.4 PKI Services support for SCEP

The Simple Certificate Enrollment Protocol (SCEP) allows you to securely issue certificates to large numbers of network devices using a primarily automatic enrollment technique. The



network devices, usually IPSEC devices such as Cisco routers, must be SCEP-enabled and preregistered (to your CA domain) before they can successfully request certificates from you. To request a certificate, the preregistered SCEP client sends a message (the certificate request) to your CA using the HTTP protocol.

Prior to z/OS V1R8, PKI Services does not support a wire protocol for receiving and fulfilling certificate requests. A Web page interface is the only means of submitting certificate requests. Due to an increasing use of certificates in routers, VPNs, and other such devices, much manual work to set up proper security had to be done.

### **z/OS V1R8 support**

Support for the Simple Certificate Enrollment Protocol is now part of PKI Services. This includes support when a device submits an encrypted and signed certificate request, and polls for a response. This support uses HTTP messages in the PKCS#7 data format. This should reduce manual administration.

You can configure PKI Services to respond automatically to some (or all) SCEP certificate requests, or to submit some (or all) SCEP certificate requests to the PKI administrator for approval or rejection. When you enable automatic enrollment, certificate requests can be automatically approved and synchronously fulfilled, based on the requestor's knowledge of a predetermined secret, the challenge passphrase.

## **6.4.1 PKI Services SCEP overview**

Figure 6-2 on page 70 shows a flow of requesting and receiving a certificate using SCEP. The administrator assigned to administer the device issues configuration commands to do the following:

- ▶ Get the SCEP CA's certificate. The configuration requires the knowledge of the CA's URL.
- ▶ To request certificates using SCEP, a SCEP requestor must be preregistered to PKI Services, your CA. You can preregister SCEP clients in batches using the `pkiprereg` utility or the PKI administrators can preregister individual SCEP clients (one client at a time) using the end-user Web pages. The PKI administrator fills out the request form by specifying the device or client name of the SCEP client, a passphrase for client authentication, and additional (optional) subject name and alternate name information. You can customize the <CONSTANT> section of the SCEP (preregistration) certificate template to supply the additional optional information. When a PKI administrator submits the form for a SCEP (preregistration) certificate request, PKI Services creates a preregistration record—not an actual certificate request—in the VSAM ObjectStore data set (requestdatabase).
- ▶ Generate and save a new key pair.
- ▶ Encrypt and sign a new certificate request (PKCSREQ).

This is encrypted under the public key of the SCEP CA. From here the certificate is either fulfilled immediately (auto-approved) or the request is queued for pending approval by the PKI administrator.

If queued, the client enters a polling loop until one of the following occurs:

- The request is cancelled by the administrator.
  - The request is rejected by the PKI administrator.
  - A timeout occurs.
- ▶ If the request is fulfilled, the client saves the certificate and uses it to conduct business.

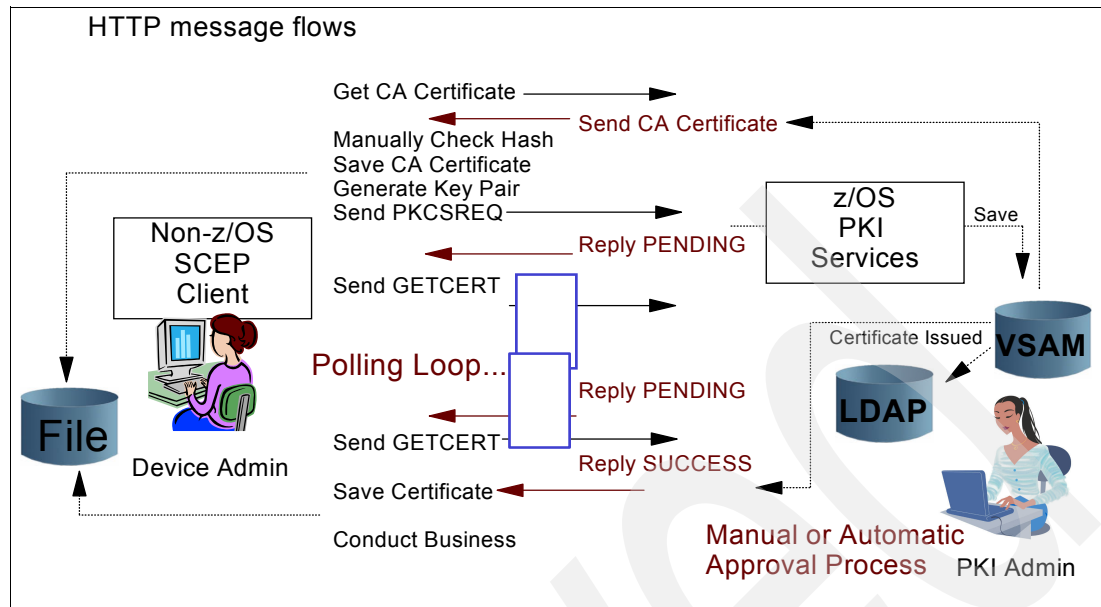


Figure 6-2 PKI Services SCEP overview

## 6.4.2 PKI Services usage considerations

When a client is preregistered, the client name is translated to lowercase characters, truncated to 32 characters if longer, and saved as the requestor to support searching of the ObjectStore. Each preregistration record must have a client name that is unique in the first 32 characters, regardless of upper or lower case.

### Preregistration process

To prevent unauthorized clients from requesting certificates, PKI Services uses a preregistration process. When a SCEP client is preregistered, it is assigned a client name, which must be part of the subjects distinguished in the certificate request that is eventually received from the client. The information that is preregistered restricts what the client can ask for in the request. While a password passphrase is not required, we would expect that most everyone would specify it. The preregistration record contains the template nickname, passphrase, and additional (optional) subject name and alternate name values.

**Note:** SCEP clients must be preregistered before requesting certificates. This is done by the PKI administrator, not the device administrator.

Each client (device) must be assigned a unique name such as a domain name, for example:

device100.tools.com

This domain name is used as the CN= or UnstructuredName= field in the certificate.

When you customize the <CONSTANT> section of the SCEP template to supply additional (optional) values for the following variables, those values are not saved in the preregistration record. However, those values are processed when the preregistered client subsequently requests a certificate.

### 6.4.3 Using PKI Services utilities

The following z/OS UNIX utility programs are shipped with PKI Services. These programs are installed in the /install-dir/pkiserv/bin directory.

<b>iclview</b>	Displays the entries in the VSAM issued certificate list (ICL) data set
<b>vosview</b>	Displays the entries contained in the VSAM ObjectStore data set (request database)
<b>pkiprereg</b>	Creates Simple Certificate Enrollment Protocol (SCEP) preregistration records

**Note:** See *z/OS Cryptographic Services PKI Services Guide and Reference*, SA22-7693 for details on how to use these utility programs.

### 6.4.4 Preregistration rules

PKI Services provides the templates to request certificates. Preregistration requires a certificate template similar to the templates PKI Services already has.

Following preregistration, when the preregistered SCEP client requests a certificate (sends a SCEP request), PKI Services searches for a preregistration record matching the client name. If found, PKI Services searches for a preregistration record matching the client name. If found, PKI Services compares the values in the request to the challenge password and any subject name or alternate name information specified by the PKI administrator or supplied in the <CONSTANT> template section. (If not found, the SCEP request is automatically rejected.)

#### New template section

<PREREGISTER>...</PREREGISTER>

This optional subsection indicates the creation of a preregistration record and contains the Simple Certificate Enrollment Protocol (SCEP) rules for approval of a SCEP request.

Example:

```
<PREREGISTER>
AuthenticatedClient=AutoApprove
SemiauthenticatedClient=AdminApprove
UnauthenticatedClient=Reject
SubsequentRequest=AutoApprove
RenewalRequest=AutoApprove
</PREREGISTER>
```

Based on the comparison of values in the request with those in the preregistration record, PKI Services considers the request to be in one of the following states:

<b>Authenticated</b>	The challenge password matches and all other preregistered values are included in the request.
<b>Semiauthenticated</b>	The challenge password matches but some other preregistered values are missing from the request.
<b>Unauthenticated</b>	The challenge password does not match or is missing.

Depending on how you customize the variables in the SCEP (preregistration) certificate template, a certificate request from an Authenticated SCEP client is either automatically approved and fulfilled synchronously or it is queued for administrator approval. Likewise, a

certificate request from an Unauthenticated or Semiauthenticated SCEP client is either queued for administrator approval or it is automatically rejected.

### **<PREREGISTER> section variables**

This section describes the valid variables you can customize in the <PREREGISTER> section of the 5-Year SCEP Certificate - Preregistration template. Some variables must be present in your <PREREGISTER> section and they are labeled as required.

- ▶ **AuthenticatedClient (required):** Specifies which action PKI Services takes when an authenticated SCEP client submits a certificate request for the first time. Valid values are:
  - **AutoApprove (default):** Automatically approves certificate requests from authenticated first-time SCEP clients and automatically creates their certificates.
  - **AdminApprove:** Submits certificate requests from authenticated first-time SCEP clients to your PKI administrator for verification and approval.
- ▶ **SemiauthenticatedClient (required):** Specifies which action PKI Services takes when a semiauthenticated SCEP client submits a certificate request for the first time. Valid values are:
  - **AdminApprove (default):** Submits certificate requests from semiauthenticated first-time SCEP clients to your PKI administrator for verification and approval.
  - **Reject:** Automatically rejects certificate requests from semiauthenticated first-time SCEP clients.
- ▶ **UnauthenticatedClient (required):** Specifies which action PKI Services takes when an unauthenticated SCEP client submits a certificate request for the first time. Valid values are:
  - **AdminApprove:** Submits certificate requests from unauthenticated first-time SCEP clients to your PKI administrator for verification and approval.
  - **Reject (default):** Automatically rejects certificate requests from unauthenticated first-time SCEP clients.
- ▶ **SubsequentRequest (optional):** Specifies which action PKI Services takes when a previously approved SCEP client submits an additional certificate request. If not set, PKI Services uses the AuthenticatedClient value. Valid values are:
  - **AutoApprove (default):** Automatically approves certificate requests from previously approved SCEP clients and automatically creates their certificates.
  - **AdminApprove:** Submits certificate requests from previously approved SCEP clients to your PKI administrator for verification and approval.
  - **Reject:** Automatically rejects SCEP requests from previously approved clients.
- ▶ **RenewalRequest (optional):** Specifies which action PKI Services takes when a previously approved SCEP client submits a certificate renewal request. If not set, PKI Services uses the AuthenticatedClient value. Valid values are:
  - **AutoApprove (default):** Automatically approves certificate renewal requests from previously approved SCEP clients and automatically creates their certificates.
  - **AdminApprove:** Submits certificate renewal requests from previously approved SCEP clients to your PKI administrator for verification and approval.
  - **Reject:** Automatically rejects certificate renewal requests from previously approved SCEP clients.

**Note:** See *z/OS Cryptographic Services PKI Services Guide and Reference*, SA22-7693 for the details on how to set up and use the template, and for the steps for starting PKI Services.

### pkiprereg utility

The pkiprereg program creates SCEP preregistration records in batch. The pkiprereg utility can be used to mass-preregister a bunch of devices. It can also be used to generate random passwords for the SCEP clients. Since it works off a z/OS UNIX file, the information can be saved and disseminated to the SCEP device administrators as needed.

**Note:** Preregistration records can be created, viewed, and deleted individually via the PKI administration Web pages, or they can be managed in bulk using the new pkiprereg z/OS UNIX utility. See *z/OS Cryptographic Services PKI Services Guide and Reference*, SA22-7693 for details on how to use the pkiprereg utility.

### SCEP messages

SCEP messages sent to PKI Services are encrypted using the CA's public key. Normal setup gives only signing capability to the CA certificate, not decryption capability. It is necessary to either create or recreate a CA certificate with additional keyUsage, or create a registration authority (RA) certificate for this purpose, signed by the CA, which is recommended.

### Establishing a CA and RA certificate

To create and sign digital certificates for others, you need to establish a CA certificate, and an optional RA certificate, and their associated private keys, using the RACDCERT command, as follows:

```
RACDCERT ID(daemon) GENCERT SUBJECTSDN(ra_dn) KEYUSAGE(HANDSHAKE)
SIGNWITH(CERTAUTH LABEL('ca_label')) NOTAFTER(DATE(ca_expires))
WITHLABEL('ra_label')

RACDCERT ID(daemon) CONNECT(LABEL('ra_label') RING(ca_ring))
```

### pkiserv.tmpl file

Create the preregistration template in the pkiserv.tmpl file by using a 5-Year SCEP certificate preregistration template as a model. Then decide what fields the PKI administrator should supply using <CONTEXT>. Also, decide what fields should be hard coded using <CONSTANT>. Customize the <PREREGISTER> section.

### pkiserv.conf file

Enable SCEP in the pkiserv.conf file. Indicate the RA certificate, if any, in the SAF section using:

```
KeyRing=PKISRVD/CARing
```

The label of the PKI Services RA certificate can be as follows:

```
RALabel=Local PKI RA
```

Turn on SCEP in the CertPolicy section and enable the Simple Certificate Enrollment Protocol to be True or False:

```
EnableSCEP=T
```

Archived

## RACF health checks

There are several ways and various tools that can examine whether your RACF environment is healthy and set up in a way your installation expects. In this chapter we describe how IBM Health Checker for z/OS can support your efforts to manage RACF.

The following topics are included:

- ▶ General information about IBM Health Checker for z/OS
- ▶ What is true for all the RACF checks
- ▶ Introduction of the new checks which were added in z/OS V1R8
- ▶ Information on the enhanced RACF checks which existed prior to z/OS V1R8

## 7.1 IBM Health Checker for z/OS

IBM Health Checker for z/OS is a z/OS base component.

Since z/OS V1R7 it is part of and delivered with z/OS, but it is also available as a download from the Web for use in previous z/OS versions.

The objective of IBM Health Checker for z/OS is to identify potential problems *before* they impact system availability or, in worst cases, cause outages. It checks the current active installation definitions and compares the values to those suggested by IBM or overridden by the installation. IBM Health Checker for z/OS produces output in the form of detailed messages to let you know of both potential problems and suggested actions to take. Individual products, z/OS components (for example, RACF), or ISV software can provide checks that take advantage of IBM Health Checker for z/OS framework.

### 7.1.1 Health checker overview

IBM Health Checker for z/OS runs in its own address space. Figure 7-1 on page 77 gives an overview of how the started task works with local check routines:

<b>Started Task</b>	Provides the services for the checks and the externals for operators and system programmers. It can be called the backbone and the primary support of the address space is to provide check routines.
<b>Check routines</b>	Check routines are the mechanisms to identify best practices, thresholds, and single points of failure. A check is owned, delivered, and supported by a specific component (for example, RACF). <i>Local checks</i> run in the IBM Health Checker for z/OS address space, <i>remote checks</i> run as tasks in the address space of the caller.
<b>Installation overrides</b>	These are the changes an installation can make if some check values are not suitable for their environment or configuration. The changes can be made permanent through statements in the HZSPRMxx parmlib member or temporary when using SDSF or the MODIFY operator command.
<b>Check output</b>	Contains messages which show the result of a check and the suggested action. You can view the messages, using SDSF, (E)JES, or the HSZPRINT utility or a log stream if the results should be archived.



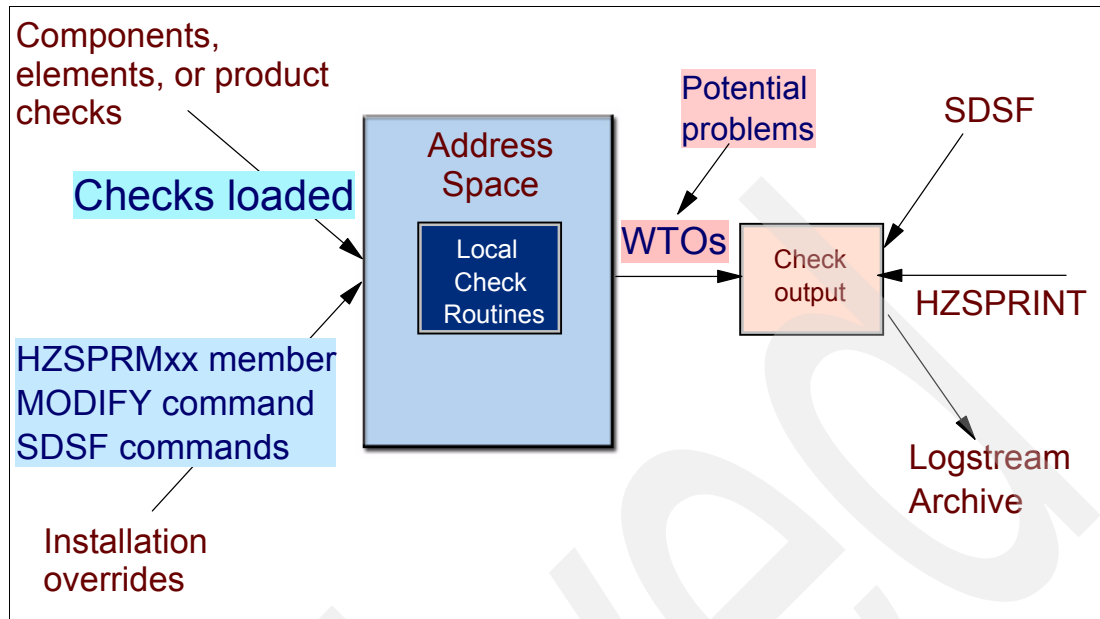


Figure 7-1 Health Checker for z/OS runs in its own address space

## Health checks

The value of IBM Health Checker for z/OS comes basically from the quality and quantity of the checks. Health checks have the following characteristics:

- ▶ A check is actually a program or routine that identifies potential problems before they impact your availability or, in worst cases, cause outages.
- ▶ A check is owned, delivered, and supported by the component, element, or product that writes it. For example, the RACF checks are the responsibility of the RACF development team.
- ▶ Checks have different severities (low, medium, high).
- ▶ Checks are small routines that:
  - Look for changes in settings or configuration values that occur dynamically over the life of an IPL.
  - Can be run periodically to keep the installation aware of changes.
  - Analyze threshold levels approaching the upper limits, especially those that might occur gradually.
  - Analyze single points of failure in a configuration.
  - Examine unhealthy combinations of configurations or values that an installation might not think to check.
- ▶ Checks can be written by IBM, by the installation, or by independent software vendors.
- ▶ Checks can be dynamically added to or deleted from a running system.
- ▶ Checks for IBM Health Checker for z/OS are delivered both as an integrated part of a z/OS release or separately, as PTFs. Many new and updated checks will be distributed as PTFs, so that they are not dependent on z/OS release boundaries and can be added at any time.

## 7.1.2 Flow of IBM Health Checker for z/OS

As illustrated in Figure 7-2 on page 79, IBM Health Checker for z/OS operates according to the following process flow:

### 1. Check values provided by components

Each check includes a set of pre-defined values, such as:

- Interval, or how often the check will run
- Severity of the check, which influences how check output is issued
- Routing and descriptor codes for the check

You can update or override some check values using either SDSF or statements in the HZSPRMxx parmlib member or the MODIFY command.

### 2. Check output

A check issues its output as write to operator (WTO) and other messages, which you can view using SDSF, the HZSPRINT utility, or a log stream that collects a history of check output. If a check finds a deviation from best practices or a potential problem, it issues a WTO message known as an *exception*. Check exception messages include not only a description of the potential problem found, including the severity, but also information on what to do to fix the potential problem.

### 3. Resolve check exceptions

To get the best results from IBM Health Checker for z/OS, you should let it run continuously on your system so that you will know when your system has changed dynamically from best practice values. When you get an exception, you should resolve it using the information in the check exception message, or override the check values so that you do not receive the same exceptions over and over. You can use either SDSF, the HZSPRMxx parmlib member, or the IBM Health Checker for z/OS **MODIFY (F hzsproc)** command to manage checks.

### 4. Rerun check

If you solve an exception by changing a product setting or system control, it is a good policy to rerun the checks related with this action to ensure that the problem identified was fixed.

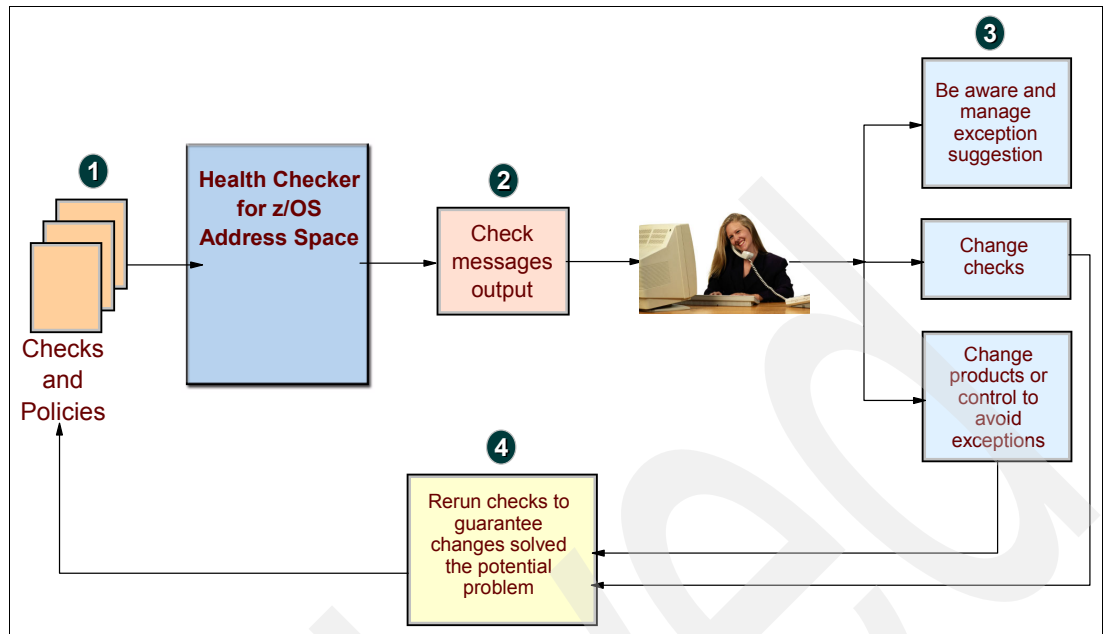


Figure 7-2 Flow of the Health Checker for z/OS

### 7.1.3 Security of IBM Health Checker for z/OS

Both IBM Health Checker for z/OS and users looking at check output need access to resources. You must create the following security definitions to control access and maintain security for these resources:

- ▶ Create a user ID for IBM Health Checker for z/OS.
- ▶ Associate the new user ID with the IBM Health Checker for z/OS started task.
- ▶ Give the IBM Health Checker for z/OS started task permission to required data sets.
- ▶ If using a log stream, define UPDATE access for the IBM Health Checker for z/OS started task to each RESOURCE (log\_stream\_name).

**Note:** You must set up security for IBM Health Checker for z/OS the same way you would do it for any other started task.

#### **Security for SDSF, (E)JES and HZSPRINT**

If you plan to use SDSF, (E)JES, or HZSPRINT to view check output, authorize SDSF, (E)JES or HZSPRINT users to QUERY and MESSAGES services via profiles in the general resource XFACILIT class.

For further information see *IBM Health Checker for z/OS User's Guide*, SA22-7994.

## 7.1.4 User interface to manage checks

You have the choice to use different interfaces to interact with IBM Health Checker for z/OS. Your exact task will determine which interface to select.

- ▶ If you want to make dynamic, temporary changes to checks, such as deactivating, adding, running, or temporarily updating check values, use:
  - SDSF panels  
CK command
  - (E)JES panels  
HC command
  - MODIFY command  
Temporary check changes, see “Health Checker for z/OS commands via MODIFY command” on page 86
- ▶ If you want to make *persistent* changes to checks that persist across check refreshes and restarts of IBM Health Checker for z/OS, use:
  - HZSPRMxx parmlib member  
Persistent check changes, see “HZSPRMxx parmlib member and policies” on page 89

## 7.1.5 Using SDSF panels

SDSF provides an interface which is easy to use. You can change the IBM Health Checker for z/OS behavior from SDSF screens, and you can manage the check’s status and results as well.

SDSF has a new CK command where you can access new panels to work with and control IBM Health Checker for z/OS.

Figure 7-6 on page 82 and Figure 7-7 on page 83 show the new SDSF panel display after issuing the CK command.

### Protecting checks on the CK panel in SDSF

You can protect the checks from IBM Health Checker for z/OS that are displayed on the CK panel. This is done by defining resource names in the XFACILIT class, as shown in Table 7-1.

Table 7-1 .Authority required to use action characters and overtypes

Action character or overtype	Function	Resource name	Class	Access
A	Activate	HZS.sysname.owner.name.ACTIVATE	XFACILIT	UPDATE
D	Display	HZS.sysname.owner.name.QUERY	XFACILIT	READ
E	Refresh	HZS.sysname.owner.name.REFRESH	XFACILIT	CONTROL
H	Deactivate	HZS.sysname.owner.name.DEACTIVATE	XFACILIT	UPDATE
P	Delete	HZS.sysname.owner.name.DELETE	XFACILIT	CONTROL
R	Run	HZS.sysname.owner.name.RUN	XFACILIT	UPDATE
S and X	Browse, Print	HZS.sysname.owner.name.MESSAGES	XFACILIT	READ
S and X	Browse, Print	HZS.sysname.owner.name.MESSAGES	XFACILIT	READ

## Messages on an authorization failure

Figure 7-3 shows the message displayed when a user without authorization attempts to use the ACTION characters shown in Figure 7-6 on page 82 on the SDSF panel display. In this example the user used the **D** action character on check CNZ\_AMRF\_EVENTUAL\_ACTION\_MSGS.

```
ICH408I USER(ROGERS ) GROUP(SYS1 ) NAME(ROGERS )
HXS.SC70.IBMCNZ.CNZ_AMRF_EVENTUAL_ACTION_MSGS.QUERY
CL(XFACILIT)
INSUFFICIENT ACCESS AUTHORITY
FROM HXS.** (G)
ACCESS INTENT(READ ) ACCESS ALLOWED(NONE )
```

Figure 7-3 RACF authorization failure message for a user ID using the D action character

## Example of protecting checks

To protect all checks and permit a user to control the checks, you can define generic profiles as follows:

```
RDEFINE XFACILIT HXS.** UACC(NONE)
PERMIT HXS.** CLASS(XFACILIT) ID(userid or groupid) ACCESS(CONTROL)
SETROPTS RACLIST(XFACILIT) REFRESH
```

After the authorization is made, the user uses the **D** action character and receives the display shown in Figure 7-4.

```
SDSF HEALTH CHECKER DISPLAY SC70                                COMMAND ISSUED
COMMAND INPUT ==>                                              SCROLL ==> PAGE
RESPONSE=SC70
HXS0200I 16.47.50 CHECK SUMMARY      508
CHECK OWNER      CHECK NAME                STATE STATUS
IBMCNZ           CNZ_AMRF_EVENTUAL_ACTION_MSGS  AE  SUCCESSFUL
  A - ACTIVE      I - INACTIVE
  E - ENABLED     D - DISABLED
  G - GLOBAL CHECK + - CHECK ERROR MESSAGES ISSUED
```

Figure 7-4 Display using D action character

## Columns on the CK display

Figure 7-5 shows some of the columns on the CK display. There are 42 columns in total. Use PF11 to display the columns not shown on the main display panel.

Column Name	Title (Displayed)	Width	Description
NAME	NAME	32	Name of the check
OWNER	CheckOwner	16	Check owner
STATE	State	18	Check state
STATUS	Status	18	Check status
RESULT	Result	6	Result code from the last invocation of the check
DIAG1	Diag1	8	Diagnostic data from check, word 1
DIAG2	Diag2	8	Diagnostic data from check, word 2 U
DIAGFROM	DiagFrom	8	Source of the diagnostic data, words 1 and 2: ABEND,HCHECKER or CHECKRTN
GLOBAL	Global	6	Indicator of whether the check is global
GLOBALSY	GlobalSys	9	Name of the system where the global check is running
EXCOUNT	ExcCount	8	Number of exceptions detected by this check on the last iteration
COUNT	RunCount	8	Number of times the check has been invoked
FAIL	Fail	4	Number of times the check failed
SEVERITY	Severity	8	Severity level of the check (HIGH,MEDIUM,LOW,NONE)

Figure 7-5 Columns that appear on the CK display

Figure 7-6 shows the display of all the health checks that can be seen when you issue the SDSF CK command.

Display Filter View Print Options Help			
-----			
SDSF HEALTH CHECKER DISPLAY SC74		LINE 1-27 (68)	
ACTION=//-Block,=-Repeat,+Extend,A-Activate,D-Display,E-Refresh,H-Deactivate,			
ACTION=P-Delete,R-Run,S-Browse,U-RemoveCat,X-Print			
COMMAND INPUT ==>		SCROLL ==> CSR	
NAME	CheckOwner	State	Status
ASM_LOCAL_SLOT_USAGE	IBMASM	ACTIVE(ENABLED)	SUCCESSFUL
ASM_NUMBER_LOCAL_DATASETS	IBMASM	ACTIVE(ENABLED)	EXCEPTION-LOW
ASM_PAGE_ADD	IBMASM	ACTIVE(ENABLED)	SUCCESSFUL
ASM_PLPA_COMMON_SIZE	IBMASM	ACTIVE(ENABLED)	SUCCESSFUL
ASM_PLPA_COMMON_USAGE	IBMASM	ACTIVE(ENABLED)	SUCCESSFUL
CNZ_AMRF_EVENTUAL_ACTION_MSGS	IBMCNZ	ACTIVE(ENABLED)	EXCEPTION-LOW
CNZ_CONSOLE_MASTERAUTH_CMDSYS	IBMCNZ	ACTIVE(ENABLED)	SUCCESSFUL
CNZ_CONSOLE_MSCOPE_AND_ROUTECD	IBMCNZ	ACTIVE(ENABLED)	EXCEPTION-LOW
CNZ_CONSOLE_ROUTECD_11	IBMCNZ	ACTIVE(ENABLED)	EXCEPTION-LOW
CNZ_EMCS_HARDCOPY_MSCOPE	IBMCNZ	ACTIVE(ENABLED)	SUCCESSFUL
CNZ_EMCS_INACTIVE_CONSOLES	IBMCNZ	ACTIVE(DISABLED)	GLOBAL-SC75
CNZ_SYSCONS_MSCOPE	IBMCNZ	ACTIVE(ENABLED)	SUCCESSFUL
CNZ_SYSCONS_PD_MODE	IBMCNZ	ACTIVE(ENABLED)	SUCCESSFUL
CNZ_SYSCONS_ROUTECD	IBMCNZ	ACTIVE(ENABLED)	SUCCESSFUL
CNZ_TASK_TABLE	IBMCNZ	ACTIVE(ENABLED)	SUCCESSFUL
CSTCP_SYSTCPIP_CTRACE_TCPIP	IBMCS	ACTIVE(ENABLED)	SUCCESSFUL
CSTCP_TCPMAXRCVBUFRSIZE_TCPIP	IBMCS	ACTIVE(ENABLED)	SUCCESSFUL
CSVAM_CSM_STG_LIMIT	IBMCS	ACTIVE(ENABLED)	SUCCESSFUL
GRS_CONVERT_RESERVES	IBMGRS	ACTIVE(DISABLED)	GLOBAL-SC75
GRS_EXIT_PERFORMANCE	IBMGRS	ACTIVE(ENABLED)	SUCCESSFUL
GRS_GRSQ_SETTING	IBMGRS	ACTIVE(ENABLED)	SUCCESSFUL
GRS_MODE	IBMGRS	ACTIVE(DISABLED)	GLOBAL-SC75
GRS_RNL_IGNORED_CONV	IBMGRS	ACTIVE(DISABLED)	GLOBAL-SC75
GRS_SYNCHRES	IBMGRS	ACTIVE(ENABLED)	SUCCESSFUL

Figure 7-6 SDSF new panel display using the CK command

NAME	CheckOwner	State	Status
RACF_FACILITY_ACTIVE	IBMRACF	ACTIVE(ENABLED)	SUCCESSFUL
RACF_GRS_RNL	IBMRACF	ACTIVE(ENABLED)	SUCCESSFUL
RACF_IBMUSER_REVOKED	IBMRACF	ACTIVE(ENABLED)	EXCEPTION-MEDIUM
RACF_OPERCMDS_ACTIVE	IBMRACF	ACTIVE(ENABLED)	SUCCESSFUL
RACF_SENSITIVE_RESOURCES	IBMRACF	ACTIVE(ENABLED)	EXCEPTION-MEDIUM
RACF_TAPEVOL_ACTIVE	IBMRACF	ACTIVE(ENABLED)	EXCEPTION-MEDIUM
RACF_TEMPDSN_ACTIVE	IBMRACF	ACTIVE(ENABLED)	EXCEPTION-MEDIUM
RACF_TSOAUTH_ACTIVE	IBMRACF	ACTIVE(ENABLED)	SUCCESSFUL
RACF_UNIXPRIV_ACTIVE	IBMRACF	ACTIVE(ENABLED)	SUCCESSFUL
RSM_AFQ	IBMRSM	ACTIVE(ENABLED)	SUCCESSFUL
RSM_HVSHARE	IBMRSM	ACTIVE(ENABLED)	SUCCESSFUL
RSM_MAXCADS	IBMRSM	ACTIVE(ENABLED)	SUCCESSFUL
RSM_MEMLIMIT	IBMRSM	ACTIVE(ENABLED)	EXCEPTION-LOW
RSM_REAL	IBMRSM	ACTIVE(ENABLED)	SUCCESSFUL
RSM_RSU	IBMRSM	ACTIVE(ENABLED)	SUCCESSFUL
SDUMP_AUTO_ALLOCATION	IBMSDUMP	ACTIVE(ENABLED)	SUCCESSFUL
SDUMP_AVAILABLE	IBMSDUMP	ACTIVE(ENABLED)	SUCCESSFUL
USS_AUTOMOUNT_DELAY	IBMUSS	ACTIVE(ENABLED)	SUCCESSFUL
USS_FILESYS_CONFIG	IBMUSS	ACTIVE(ENABLED)	EXCEPTION-MEDIUM
USS_MAXSOCKETS_MAXFILEPROC	IBMUSS	ACTIVE(ENABLED)	EXCEPTION-LOW
VSAM_INDEX_TRAP	IBMVSAM	ACTIVE(ENABLED)	EXCEPTION-MEDIUM
VSM_ALLOWUSERKEYCSA	IBMVSM	INACTIVE(ENABLED)	INACTIVE
VSM_CSA_CHANGE	IBMVSM	ACTIVE(ENABLED)	SUCCESSFUL
VSM_CSA_LIMIT	IBMVSM	ACTIVE(ENABLED)	SUCCESSFUL
VSM_CSA_THRESHOLD	IBMVSM	ACTIVE(ENABLED)	SUCCESSFUL
VSM_PVT_LIMIT	IBMVSM	ACTIVE(ENABLED)	SUCCESSFUL
VSM_SQA_LIMIT	IBMVSM	ACTIVE(ENABLED)	SUCCESSFUL
VSM_SQA_THRESHOLD	IBMVSM	ACTIVE(ENABLED)	SUCCESSFUL
XCF_CDS_SEPARATION	IBMXCF	ACTIVE(ENABLED)	EXCEPTION-HIGH
XCF_CF_CONNECTIVITY	IBMXCF	ACTIVE(ENABLED)	EXCEPTION-MEDIUM
XCF_CF_STR_EXCLLIST	IBMXCF	ACTIVE(ENABLED)	SUCCESSFUL
XCF_CF_STR_PREFLIST	IBMXCF	ACTIVE(ENABLED)	EXCEPTION-MEDIUM
XCF_CLEANUP_VALUE	IBMXCF	ACTIVE(ENABLED)	SUCCESSFUL
XCF_DEFAULT_MAXMSG	IBMXCF	ACTIVE(ENABLED)	SUCCESSFUL
XCF_FDI	IBMXCF	ACTIVE(ENABLED)	SUCCESSFUL
XCF_MAXMSG_NUMBUF_RATIO	IBMXCF	ACTIVE(ENABLED)	SUCCESSFUL
XCF_SFM_ACTIVE	IBMXCF	ACTIVE(ENABLED)	EXCEPTION-MEDIUM
XCF_SIG_CONNECTIVITY	IBMXCF	ACTIVE(ENABLED)	SUCCESSFUL
XCF_SIG_PATH_SEPARATION	IBMXCF	ACTIVE(ENABLED)	SUCCESSFUL
XCF_SIG_STR_SIZE	IBMXCF	ACTIVE(ENABLED)	SUCCESSFUL
XCF_SYSPLEX_CDS_CAPACITY	IBMXCF	ACTIVE(ENABLED)	SUCCESSFUL
XCF_TCLASS_CLASSLEN	IBMXCF	ACTIVE(ENABLED)	SUCCESSFUL
XCF_TCLASS_CONNECTIVITY	IBMXCF	ACTIVE(ENABLED)	SUCCESSFUL
XCF_TCLASS_HAS_UNDESIG	IBMXCF	ACTIVE(ENABLED)	SUCCESSFUL

Figure 7-7 SDSF new panel display using CK command

## SDSF commands to customize panels

The command SET ACTION permits you to see all available commands in the top panel:

```
SET ACTION (ON|LONG|SHORT|OFF|?)
```

To display the action character, use the following command:

```
SET ACTION ON
```

You can customize how your main panel is displayed by choosing which columns to show first.

There are many columns with information about each check. You can choose what to see in the first screen displayed. The command **ARRANGE** allows you to place columns in the order that you want, and select the width of a column. The **ARRANGE** command can be used as follows:

```
ARRANGE from column After|Before to column
ARRANGE from-column First|Last|width
```

Figure 7-8 shows the SDSF CK command panel after the **ARRANGE** command has changed the order of the columns. The two columns that are now displayed are:

**Result**            Result code from the last invocation of the check

**RunCount**        Number of times the check has been invoked

Display Filter View Print Options Help

-----

SDSF HEALTH CHECKER DISPLAY SC70

LINE 1-23 (53)

COMMAND INPUT ==>

SCROLL ==> CSR

ACTION=//-Block,=-Repeat,+-Extend,A-Activate,D-Display,E-Refresh,H-Deactivate,  
ACTION=P-Delete,R-Run,S-Browse,U-RemoveCat,X-Print

NP	NAME	CheckOwner	Status	Result	RunCount
	CNZ_AMRF_EVENTUAL_ACTION_MSGS	IBMCNZ	SUCCESSFUL	0	2
	CNZ_CONSOLE_MASTERAUTH_CMDSYS	IBMCNZ	SUCCESSFUL	0	5
	CNZ_CONSOLE_MSCOPE_AND_ROUTCODE	IBMCNZ	EXCEPTION-LOW	4	5
	CNZ_CONSOLE_ROUTCODE_11	IBMCNZ	EXCEPTION-LOW	4	5
	CNZ_EMCS_HARDCOPY_MSCOPE	IBMCNZ	SUCCESSFUL	0	5
	CNZ_EMCS_INACTIVE_CONSOLES	IBMCNZ	SUCCESSFUL	0	5
	CNZ_SYSCONS_MASTER	IBMCNZ	SUCCESSFUL	0	5
	CNZ_SYSCONS_MSCOPE	IBMCNZ	SUCCESSFUL	0	5
	CNZ_SYSCONS_PD_MODE	IBMCNZ	SUCCESSFUL	0	97
	CNZ_SYSCONS_ROUTCODE	IBMCNZ	SUCCESSFUL	0	5
	CNZ_TASK_TABLE	IBMCNZ	SUCCESSFUL	0	386
	GRS_CONVERT_RESERVES	IBMGRS	EXCEPTION-LOW	4	1
	GRS_EXIT_PERFORMANCE	IBMGRS	EXCEPTION-LOW	4	5
	GRS_MODE	IBMGRS	SUCCESSFUL	0	1
	GRS_SYNCHRES	IBMGRS	SUCCESSFUL	0	97

Figure 7-8 CK display panel with the order of the columns changed

## SDSF and the MODIFY command

SDSF uses a subset of the **MODIFY** command. This means that some commands can be used from SDSF, and others cannot. Table 7-2 on page 85 shows differences between SDSF and the **MODIFY** command.

It is possible to change some values dynamically by changing them directly on the SDSF screen. For example, you can change the check's severity from **LOW** to **MEDIUM**, just by overtyping on the SDSF screen if you are authorized. When you do this, SDSF issues a **MODIFY** command in the background for you, changing the severity to **MEDIUM**.



Table 7-2 Issuing commands with SDSF or the MODIFY command

Function	SDSF	MODIFY command: f hzsproc parameters
<ul style="list-style-type: none"> <li>▶ Change check states</li> <li>▶ Add new checks</li> <li>▶ Delete checks</li> <li>▶ Display check information</li> <li>▶ Run checks</li> <li>▶ Change interval to run each check</li> <li>▶ Update or override check values in use</li> <li>▶ Categorize checks</li> </ul>	Yes	Yes
Stop IBM Health Checker for z/OS address space	No	Yes
Connect to IBM Health Checker for z/OS log stream	No	Yes
Request processing of HZSPRMxx parmlib members	No	Yes
Policy statement support	No	Yes - See Note

**Note:** Policy statements are allowed in the MODIFY command, but we recommend this only for test purposes. When you need permanent changes, or permanent overrides, you must use the HZSPRMxx parmlib member to create policies with the changes you want.

### SDSF compatibility and requirements

Every release of z/OS SDSF requires the level of the BCP that it ships with.

In the case of an installation running on a sysplex, you may need WebSphere MQ. Without the MQ support, SDSF shows checks only for the system you are logged on to. Sysplex-wide data requires WebSphere MQ on each system.

## 7.1.6 Using (E)JES panels

(E)JES provides a similar interface for JES3 users, which is also easy to use.

The HC command provides you access new to the panels to work with and control IBM Health Checker for z/OS.

Figure 7-9 on page 86 shows an example of (E)JES panel display after issuing the HC command.



## MODIFY command

You can update or override some check values using the MODIFY command. These are called installation updates. The commands are useful for making dynamic, temporary changes to checks. You might do this if some check values are not suitable for your environment or configuration. The MODIFY command has the parameter options shown in Table 7-3. The format of the command is:

```
F HZSPROC,parameters
```

Table 7-3 MODIFY command subcommands

MODIFY parameter	Purpose of the parameter
ACTIVATE	Sets the check state to active.
ADD/ADDREPLACE/ REMOVE POLICY	Adds, replaces, or removes a policy statement.
ADDNEW	Adds new checks to Health Check.
ADD/REPLACE/SET	Used to indicate which parmlib members' suffixes are going to be in use with Health Checker.
DEACTIVATE	Disables running of specified check.
DELETE	Deletes the specified check from Health Check. Once it is deleted you can bring it back only by refresh processing.
DELETE,FORCE	Deletes a check that is running.
DISPLAY	Shows information about the check.
DISPLAY,CHECKS	Finds the check owner and check name.
LOGGER	Connects to a pre-defined log stream.
REFRESH	Deletes the check, than performs the ADDNEW function, adding the check to Health Checker again.
RUN	Run a check immediately.
STOP	Stops the Health Checker.
UPDATE	Allows a temporary update to the current default or override values for a specific check. The new value is in effect until the next refresh for the specified check.

## Command example

Figure 7-10 shows use of the MODIFY command to display check settings.

```
F hzsproc,DISPLAY,CHECKS(check_owner,check_name),  
  [SUMMARY|DETAIL]  
  [,ANY|,NOTDELETED|,DELETED]  
  [,POLICYEXCEPTIONS][,EXCEPTIONS]
```

Figure 7-10 Command to display check settings

Figure 7-11 on page 88 is an example of the details about a specific check.

```

F HZSPROC,DISPLAY,CHECKS,CHECK=(IBMRACF,RACF_IBMUSER_REVOKED),DETAIL
HZS0201I 10.49.38 CHECK DETAIL      072
CHECK(IBMRACF,RACF_IBMUSER_REVOKED)
STATE: ACTIVE(ENABLED)              STATUS: EXCEPTION-MED
EXITRTN: IRRHCA00
LAST RAN: 07/06/2006 10:45      NEXT SCHEDULED: 07/07/2006 10:45
INTERVAL: 24:00
EXCEPTION INTERVAL: SYSTEM
SEVERITY: MEDIUM
WTOTYPE: EVENTUAL ACTION
SYSTEM DESC CODE: 3
THERE ARE NO PARAMETERS FOR THIS CHECK
REASON FOR CHECK: IBMUSER should be revoked.
MODIFIED BY: N/A
DEFAULT DATE: 20051111
ORIGIN: HZSADDCK
LOCALE: HZSPROC
DEBUG MODE: OFF  VERBOSE MODE: NO

```

Figure 7-11 *MODIFY command example*

## Temporary check changes

Temporary check changes are useful when you need to test a modification before making it permanent, or when you just need to change a check for a specific situation. A temporary change is in effect until the first check refresh or system IPL. When you want a change to be permanent you must use the HZSPRMxx parmlib member.

Use the SDSF panel or the following MODIFY command to make temporary check changes:

```
f hzsproc,UPDATE,filters,action
```

In this example, *filters* can be:

```

CHECK=(check_owner,check_name)
EXITRTN=exit routine
CATEGORY=( [{ ONLY | ANY | EVERY | EXCEPT }, ] [category1 [, ..., categoryn]] )

```

Filters specify which check or checks you wish to take an action against. You can specify wildcard characters \* and ? for filters. An asterisk (\*) represents any string having a length of zero or more characters. A question mark (?) represents a position that may contain any single character.

Also in this example, the update *action* can be:

```

[ , ACTIVE | INACTIVE ]
[ , ADDCAT=(cat1 [, ..., cat16]) ]
[ , DATE=date ]
[ , DEBUG={ OFF | ON } ]
[ , DESC CODE=(desc code1 [, ..., desc code n]) ]
[ , { INTERVAL=ONETIME | INTERVAL=hhh:mm } ]
[ , PARM=parameter, REASON=reason, DATE=date ]
[ , REASON=reason ]
[ , REPCAT=(cat1 [, cat2 [, ..., cat16]] ) ]
[ , REMCAT=(cat1 [, cat2 [, ..., cat16]] ) ]
[ , ROUT CODE=(rout code1 [, ..., rout code n]) ]
[ , SEVERITY={ HIGH | MEDIUM | LOW | NONE } ]
[ , WTOTYPE={ CRITICAL | EVENTUAL | INFORMATIONAL | HARDCOPY | NONE } ]

```

## 7.1.8 HZSPRMxx parmlib member and policies

Each time checks are refreshed or added, or when there is an IPL, IBM Health Checker processes policy information from the HZSPRMxx parmlib members in use. The system applies these policy statements in the order they occur in the HZSPRMxx parmlib members. Define an HZSPRMxx parmlib member to define and modify policies, and to enable log stream processing.

### Defining a policy

You can make persistent changes by creating policies in the HZSPRMxx parmlib member. The IBM Health Checker for z/OS policy simply consists of the following:

- ▶ A set of update statements in the HZSPRMxx parmlib member or members currently in use for a system.
- ▶ The information in your IBM Health Checker for z/OS policy is applied to all existing checks and to any new checks you add.
- ▶ IBM Health Checker for z/OS processes policy information every time checks are refreshed or added, or when there is an IPL.
- ▶ The policy is the place to put any check changes you want to make persistent and to have applied to any checks you add in the future.
- ▶ Starting with z/OS V1R8, you can create multiple policies and switch between them. (Systems at the z/OS V1R4 through R7 level with IBM Health Checker for z/OS support installed can have only one policy per system.)
- ▶ Use the non-policy statements to test changing values.

### HZSPRMxx parmlib member policy

A policy allows you to make permanent overrides or changes in the IBM Health Checker for z/OS behavior. Before creating a policy, test your changes using the following MODIFY command:

```
f hzsproc,UPDATE,filters,action
```

When you are sure about the changes you want, change your HZSPRMxx parmlib member by adding a new policy. Including other non-policy statements in your HZSPRMxx parmlib member will be ineffective, because the parmlib member specified in the HZSPROC procedure is processed before any checks are added or the Health Checker for z/OS begins running.

## 7.1.9 Policy statements

Use the policy statement in the HZSPRMxx parmlib member to establish permanent overrides to existing checks. Policy statements are as follows:

- ▶ ADD POLICY
- ▶ ADDREPLACE POLICY
- ▶ REMOVE POLICY

They are applied immediately, and are applied again whenever a check is added or refreshed. We recommend that you use policy statements only in an HZSPRMxx parmlib member.

All policy statements are named for easy reference, and each one has a date and reason as well. When a check is updated, a policy with an older date may not be applied, forcing a review of policy statements when a check is updated.

You can use a policy statement to apply any kind of update command to change checks or to permanently delete checks. You can change or create a policy, changing HZSPRMxx or creating a new HZSPRMxx parmlib member and concatenating it.

Use the following policy statement to create, replace, or remove a policy:

```
{ADD | ADDREPLACE}
POLICY STATEMENTNAME(name) UPDATE(filters) [UPDATE options] REASON(reason) DATE(date)
POLICY STMT(name) DELETE(filters) REASON(reason) DATE(date)

REMOVE POLICY STATEMENT(name)
```

In this example, the *filters* can be:

```
CHECK=(check_owner,check_name)
EXITRTN=exit routine CATEGORY=([{ONLY|ANY|EVERY|EXCEPT},][category1[,...,categoryn]])
```

## Policy filters

Filters specify which check or checks you wish to take an action against. You can specify wildcard characters \* and ? for filters, as follows:

Asterisk (\*) Represents any string having a length of zero or more characters  
Question mark (?) Represents a position which may contain any single character

Filter parameters are as follows:

- ▶ CHECK is a required filter, except for the D CHECKS, filters command.
- ▶ EXITRTN=exit routine: EXITRTN specifies the HZSADDCHECK exit routine that added checks to IBM Health Checker for z/OS.
- ▶ CATEGORY=([{ONLY|ANY|EVERY|EXCEPT},][category1[,...,categoryn]]). Filter checks are by user-defined categories. The CATEGORY filters can be one of the following:
 

ONLY	These checks are in every one of the specified categories, and have only as many categories as are specified. For example, a check assigned to three categories would not match if the CATEGORY=ONLY statement on this MODIFY command specified two categories.
ANY	These checks are in any of the specified categories.
EVERY	These checks are in every specified category.
EXCEPT	Checks that are not in any of the specified categories.

## Update options

The syntax of the update options you can use to update check values is as follows:

```
UPDATE, filters
[,ACTIVE|INACTIVE]
[,ADDCAT=(cat1[,...,cat16])]
[,DATE={date | (date,NOCHECK)}]
[,DEBUG={OFF|ON}] [,VERBOSE={NO|YES}]
[,DESCCODE=(desccode1[,...,descoden])]
[,INTERVAL={ONETIME|hhh:mm}]
[,EINTERVAL={SYSTEM|HALF|hhh:mm}]
[,PARM=parameter,REASON=reason,DATE={date | (date,NOCHECK)}] [,REASON=reason]
[,REPCAT=(cat1[,cat2[,...,cat16]])]
[,REMCAT=(cat1[,cat2[,...,cat16]])]
[,ROUTCODE=(routcode1[,...,routcoden])]
```

```
[,SEVERITY={HIGH|MEDIUM|LOW|NONE}]  
[,WTOTYPE={CRITICAL|EVENTUAL|INFORMATIONAL|HARDCOPY|NONE}]
```

In the update filters, identify checks you want to change.

### Policy statement example

In the following example, we create a policy statement called `policy1` to change the `GRS_SYNCHRES` check's running interval from 1 hour to 30 minutes:

```
ADD POLICY STMT(POLICY1) UPDATE CHECK(IBMGRS,GRS_SYNCHRES)  
INTERVAL(00:30) REASON('CHANGING INTERVAL FROM 1H TO 30M')  
DATE(20050510)
```

### MODIFY command to add a policy example

You can use this statement as a test proposal with the `MODIFY` command also. You can specify filters and the update option shown in Figure 7-12 on page 91. The same options were described in the `HZSPRMxx` parmlib member in the previous examples.

```
F hzsproc,{ADD | ADDREPLACE},POLICY,STMT=stmtntname,UPDATE,filters  
[,update options],REASON=reason,DATE=date
```

Figure 7-12 *MODIFY policy command*

### Policy statement to create policy1 example

In the following example, a policy statement called `policy1` is created to change the `GRS_SYNCHRES` check's running interval from 1 hour to 30 minutes:

```
ADD POLICY STMT(POLICY1) UPDATE CHECK(IBMGRS,GRS_SYNCHRES)INTERVAL(00:30)  
REASON('CHANGING INTERVAL FROM 1H TO 30M') DATE(20050510)
```

Using the statement on the parmlib member makes it permanent. If you issue after a `MODIFY UPDATE` command (`F hcproc,UPDATE`), the updated value is valid until the next check refresh or system IPL.

It is possible to use the `MODIFY UPDATE` command directly on the parmlib member also, but this update is lost when the first check refresh is done.

**Tip:** For tests or temporary changes, use SDSF panels and the `MODIFY` command very carefully. For permanent changes, the best practice is to use the `POLICY` statement in the `HZSPRMxx` parmlib member.

### ADDREPLACE POLICY example

These same values will be applied to all checks owned by `IBMGRS`, every time they are refreshed or added. This means that all new and existing `IBMGRS` checks will be set to `HIGH` severity until the system is IPLed or IBM Health Checker for z/OS is restarted. At IPL time, you lose your policy updates unless you have updated `HZSPRMxx` parmlib member with the new (or changed) policy `P2`.

```
ADDREPLACE POLICY STMT(p2) UPDATE CHECK(ibmgrs,*) SEVERITY(high) REASON('change  
policy') DATE(20050901)
```

Figure 7-13 *Another policy example*

Use this command as an example of an `HZSPRMxx` parmlib member.

## Syntax for HZSPRMxx parmlib members

The syntax for HZSPRMxx parmlib members and the MODIFY command are similar. You can use the same parameters in both the HZSPRMxx parmlib member and the F hzsproc,parameters command, but there are differences. To specify parameters in an HZSPRMxx parmlib member, consider the following:

- ▶ Use parentheses where an equal sign is used in the MODIFY command.
- ▶ Separate parameters with blanks instead of commas.

### 7.1.10 Categories to manage and display information

IBM Health Checker for z/OS offers you a new resource, called *category*, to control your checks. When you have many checks, you can use categories to make it easier to manage or display information. Use the ADDCAT, REPCAT, and REMCAT parameters, as follows:

<b>ADDCAT</b>	Lets you add the specified check to a category
<b>REPCAT</b>	Lets you replace a category for a check
<b>REMCAT</b>	Lets you remove a check from a category

**Note:** All categories are user-defined. IBM does not define any categories for checks.

The following examples show how you can use categories in the HZSPRMxx parmlib member and in the MODIFY command to manage checks.

#### Category filters

Use the CATEGORY filter to filter actions against checks by category. For example, you might put checks into categories such as shift and offshift, global, or exception. Then you can perform actions such as activate, deactivate, or run a group of checks with one command.

It is very easy to create or change a check's category using the SDSF CK panel. However, this kind of change is temporary, as discussed previously.

#### Command example

In the example in Figure 7-14, we grouped GRS\_MODE and GRS\_SYNCHRES checks into a GRS category. The easiest way to create your own categories is on the SDSF panel, but if you want a category to be permanent, use the policy statement in the HZSPRMxx parmlib member instead. Now, it is possible to display details for this group of checks defined by the category.



```

F HZSPROC,DISPLAY,CHECKS,CATEGORY=(GRS),DETAIL
HZS0201I 10.45.01 CHECK DETAIL      725
CHECK(IBMGRS,GRS_SYNCHRES)
STATE: ACTIVE(ENABLED)                STATUS: SUCCESSFUL
EXITRTN: ISGHCADC
LAST RAN: 05/09/2005 09:54    NEXT SCHEDULED: 05/09/2005 10:54
INTERVAL: 1:00    SEVERITY: LOW    WTOTYPE: INFORMATIONAL
SYSTEM DESC CODE: 12
THERE ARE NO PARAMETERS FOR THIS CHECK
REASON FOR CHECK:  GRS synchronous RESERVE processing should be
                    enabled to avoid deadlock conditions.
MODIFIED BY: MODIFY COMMAND
CATEGORIES: GRS
DEFAULT DATE: 20050105                DEBUG MODE: OFF

CHECK(IBMGRS,GRS_MODE)
STATE: ACTIVE(ENABLED)    GLOBAL STATUS: SUCCESSFUL
EXITRTN: ISGHCADC
LAST RAN: 05/06/2005 09:54    NEXT SCHEDULED: (NOT SCHEDULED)
INTERVAL: ONETIME    SEVERITY: LOW    WTOTYPE: INFORMATIONAL
SYSTEM DESC CODE: 12
DEFAULT PARAMETERS:    STAR
REASON FOR CHECK:  GRS should run in STAR mode to improve
                    performance.
MODIFIED BY: MODIFY COMMAND
CATEGORIES: GRS
DEFAULT DATE: 20050105                DEBUG MODE: OFF

```

Figure 7-14 Command displaying a user defined category

### 7.1.11 Criteria for the checks

Now we take a closer look at the contents of a check. All checks (not only RACF checks) are identified by different criteria. The most important are:

<b>name</b>	The <i>name</i> of a check consists of the component and a further description. The RACF check names all start with RACF, for example, RACF_GRS_RNL.
<b>checkowner</b>	Every check is <i>owned</i> by a software component. The owner of the RACF checks is IBMRACF.
<b>severity</b>	The checks are classified into three categories: <i>high</i> , <i>medium</i> , and <i>low</i> according to their importance and impact.
<b>WTOtype</b>	If an exception is identified, a WTO identifying the exception is issued. Each check has one of the following defined WTOtypes: <ul style="list-style-type: none"> <li>INFORMATIONAL For information messages with a low severity. Indicates that the check found a problem that will not impact the system immediately, but that should be investigated.</li> <li>EVENTUAL For eventual action messages. Indicates that the check found a medium severity problem in the installation.</li> <li>CRITICAL For critical eventual action messages. Indicates that the check routine found a high-severity problem in the installation.</li> </ul>

	<b>HARDCOPY</b>	For hardcopy messages. Specifies that the system issues the message to the hardcopy log. This is the default if SEVERITY(NONE) is specified.
<b>state</b>		The <i>state</i> indicates whether a check will run at the next specified interval; normally it will be <i>active(enabled)</i> . But there might be situations, where the system decides to <i>disable</i> a check or you decide to <i>inactive</i> a check, so that it is not longer eligible to run.
<b>status</b>		Describes the output of the check when it last ran. If your system matches the values IBM (or the ISVs or you) specified to be best practice, the status of a check is successful. If a check determines that a situation exists on a system contrary to a recommendation, it becomes an <i>exception</i> . The exception is classified according the severity and the status is <i>exception-low</i> , <i>exception-medium</i> , or <i>exception-high</i> .
<b>interval</b>		The <i>interval</i> specifies how often a check will be executed. You can specify one time (the check will be executed only once) or you can specify an interval, for example, 24:00 signifies the check will be executed every 24 hours, meaning once a day.
<b>reason</b>		The <i>reason</i> gives background about the check and describes what the check validates.

## 7.2 Common features of all RACF checks

There are some things which are common to all RACF checks:

- ▶ The owner of the checks is IBMRACF.
- ▶ The names of the RACF checks start with RACF.
- ▶ All of the RACF checks are implemented in one module (IRRHCR00):
  - Entry codes are used to select a specific check.
  - Some checks allow or expect parameters.
- ▶ The registration module (IRRHCA00) is still driven during RACF initialization.
- ▶ All messages are in one message module (IRRHCM00).

You can find the documentation of these checks in *IBM Health Checker for z/OS User's Guide*, SA22-7994.

## 7.3 New RACF checks

As mentioned previously, the number of available RACF checks will increase over time. With z/OS 1.8 RACF introduced seven new checks. Six of them check whether a specific general resource class is activated and one checks if the IBMUSER is revoked. The following sections describe these new checks.

### 7.3.1 Check RACF\_<class-name>\_ACTIVE

There are six new checks that examine whether a single general resource class is activated. These six checks use a common check routine. The names of the checks are:

- ▶ RACF\_FACILITY\_ACTIVE

- ▶ RACF\_OPERCMDS\_ACTIVE
- ▶ RACF\_TAPEVOL\_ACTIVE
- ▶ RACF\_TEMPDSN\_ACTIVE
- ▶ RACF\_TSOAUTH\_ACTIVE
- ▶ RACF\_UNIXPRIV\_ACTIVE

This is done by a call of the macro RACROUTE REQUEST=STAT with the class name. The SAF and RACF return codes are examined.

The owner of all these checks is IBMRACF because all these checks are owned by the RACF component.

**Note:** IBM recommends that this entire baseline group of RACF resource classes be active; this is why the checks were introduced.

Table 7-4 shows the criteria of the new RACF checks.

*Table 7-4 Main criteria of the RACF\_classname\_ACTIVE checks*

Criteria	Value
OWNER	IBMRACF
Severity	Medium
WTOtype	Eventual
Interval	24:00
Reason	IBM recommends activating this class
Debug mode	Off

## Check output

As mentioned in “Using SDSF panels” on page 80, SDSF provides an easy way to look at the result of the checks. The SDSF line command **s** (for browse) displays the result of a check.

In the following discussion we show two examples, one of a successful check and one that found an exception.

## Successful check

Figure 7-15 shows the output of a successful check, meaning that the checked configuration agrees with, in this case, the IBM recommendation to have the class TSOAUTH activated.

```

HZS1098I  CHECK(IBMRA CF,RACF_TSOAUTH_ACTIVE)
HZS1090I  START TIME: 07/05/2006 15:31:38.248026
HZS1095I  CHECK DATE: 20051111  CHECK SEVERITY: MEDIUM
HZS1097I  CHECK PARM: TSOAUTH
HZS1097I

IRRH228I  IRRH228I The class TSOAUTH is active.

HZS1091I  END TIME: 07/05/2006 15:31:38.248144  STATUS: SUCCESSFUL

```

*Figure 7-15 Output of a successful check on class TSOAUTH*

## Check with an exception

Figure 7-16 shows the output of a check which results in an exception.

The check RACF\_TAPEVOL\_ACTIVE examines whether class TAPEVOL is active on the system. IBM recommends activating this class, but this installation has not activated it.

```
CHECK(IBMRA CF,RACF_TAPEVOL_ACTIVE)
START TIME: 07/05/2006 12:17:06.071478
CHECK DATE: 20051111  CHECK SEVERITY: MEDIUM
CHECK PARM: TAPEVOL

* Medium Severity Exception *

IRRH229E The class TAPEVOL is not active.

Explanation: The class is not active. IBM recommends that the
security administrator at your installation activate this class and
define in it the profiles to properly protect your system.

System Action: The check continues processing. There is no effect on
the system.

Operator Response: Report this problem to the system security
administrator and the system auditor.

System Programmer Response: None.

Problem Determination: See the RACF Auditor's Guide and the RACF
Systems Programmer's Guide.

Source:
  RACF System Programmer's Guide
  RACF Auditor's Guide

Reference Documentation:
  RACF System Programmer's Guide
  RACF Auditor's Guide

Automation: None.

Check Reason: IBM recommends activating this class

END TIME: 07/05/2006 12:17:06.072424  STATUS: EXCEPTION-MED
```

Figure 7-16 Output of a check for class TAPEVOL which finds an exception

In addition to an explanation of the exception criteria, the check output displays the date and time of the last run of the check.

**Note:** The output shows only the result from the *last* run, not all the results from previous runs. If you are interested in the results of previous runs you should set up a log stream.

## Save the check results using system logger

IBM Health Checker for z/OS retains only the check results from the last iteration of a check in the message buffer. If you want to retain a historical record of check results, which is a good idea, you must define and connect to a log stream. When you have a log stream connected, the system writes check results to the log stream every time a check completes.

### 7.3.2 Check RACF\_IBMUSER\_REVOKED

With z/OS 1.8 IBM Health Checker for z/OS introduces one further check, which verifies whether the IBMUSER user ID is still ready to be used. To prevent a defined RACF user ID from being used you should revoke it. This check verifies that the IBMUSER user ID is revoked.

It is done by a call of the macro RACROUTE REQUEST=VERIFY for the IBMUSER. The SAF and RACF return codes are examined.

**Note:** This check is a best practice recommendation from IBM because the IBMUSER user ID is intended for use only during the initial installation process. After installation, the IBMUSER user ID should be revoked so that it cannot be used by unauthorized users.

Table 7-5 shows the main criteria IBM defined for this check.

Table 7-5 Main criteria for the RACF\_IBMUSER\_REVOKED check

Criteria	Value
OWNER	IBMRACF
Severity	Medium
WTOtype	Eventual
Interval	24:00
Reason	IBMUSER should be revoked
Debug mode	Off

Other than the mentioned reason, the criteria for this check are the same as for the RACF\_<class-name>\_ACTIVE checks identified in Table 7-4 on page 95.

Figure 7-17 on page 98 is an example of the output if the RACF\_IBMUSER\_REVOKED check finds an exception.

```

CHECK(IBMRAF,RACF_IBMUSER_REVOKED)
START TIME: 07/05/2006 17:37:23.183723
CHECK DATE: 20051111  CHECK SEVERITY: MEDIUM

* Medium Severity Exception *

IRRH225E The user ID IBMUSER is not revoked.

Explanation: The user ID IBMUSER has not been revoked. IBM recommends
revoking IBMUSER.

System Action: The check continues processing. There is no effect on
the system.

Operator Response: Report this problem to the system security
administrator and the system auditor.

System Programmer Response: Revoke IBMUSER.

Problem Determination: See the RACF Auditor's Guide and the RACF
Systems Programmer's Guide.

Source:
  RACF System Programmer's Guide
  RACF Auditor's Guide
Reference Documentation:
  RACF System Programmer's Guide
  RACF Auditor's Guide

Automation: None.

Check Reason: IBMUSER should be revoked.

END TIME: 07/05/2006 17:37:23.190112  STATUS: EXCEPTION-MED

```

Figure 7-17 Check output for an exception on RACF\_IBMUSER\_REVOKED

## 7.4 Enhanced RACF checks

Two RACF checks that were available in previous releases have been modified and enhanced in z/OS V1R8.

### 7.4.1 Check RACF\_GRS\_RNL

This check evaluates whether the RACF ENQ names are in either the installation system exclusion resource name list (SERNL) or the system inclusion resource name list (SIRNL). This check verifies that no RACF ENQ names were found in the GRS Resource Name List.

The RACF service team has debugged several customer problems and outages and found that the problem or outage was caused by a customer's resource name list (RNL) changing the scope of a RACF serialization request. With z/OS V1R6, GRS introduced an enhanced ISGQUERY service that allows an application to specify the QNAME and RNAME of an ENQ and determine if the ENQ name is on an RNL.

When it runs, the RACF\_GRS\_RNL check calls the GRS ISGQUERY service for each of the ENQ names documented in *IBM Health Checker for z/OS User's Guide*, SA22-7994. If one or more ENQs are on an RNL that affects the scope of the ENQ, then the RACF\_GRS\_RNL check identifies the ENQs that have their scope changed.

**Note:** IBM recommends that installations do not convert RACF SYSTEM ENQs to SYSTEM ENQs because this can corrupt the RACF database and result in an outage.

Table 7-6 shows the criteria IBM defined for this check.

Table 7-6 Main criteria for the RACF\_GRS\_RNL check

Criteria	Value
OWNER	IBMRACF
Severity	High
WTotype	Critical
Interval	08:00
Reason	None of the RACF ENQ names should be in RNLs
Debug mode	Off

Compared to the described RACF checks in “New RACF checks” on page 94, this check has a *higher* severity and if the check finds an exception it will result in a *critical* action message. Furthermore, the check runs every 08 hours, not only once e day.

### Successful check

Figure 7-18 shows the result of a successful check.

```

CHECK(IBMRA CF,RACF_GRS_RNL)
START TIME: 07/06/2006 12:55:28.097364
CHECK DATE: 20040703  CHECK SEVERITY: HIGH

                                RACF_GRS_RNL Report

S Major   Minor                Type  QName    Rname                Type
-----
IRRH203I No RACF ENQ names were found in the GRS Resource Name List.

END TIME: 07/06/2006 12:55:28.099444  STATUS: SUCCESSFUL

```

Figure 7-18 Output of a successful check for RACF\_GRS\_RNL

If you want to know for which ENQ names the check calls the GRS ISGQUERY, you can modify the check and set the debug mode (or verbose mode) *on*. This can be done via the different interfaces mentioned in “User interface to manage checks” on page 80. The following example shows how it is done using the MODIFY command:

**F HZSPROC,UPDATE,CHECK=(IBMRA CF,RACF\_GRS\_RNL),DEBUG=ON**

If you rerun the check (for example, through the action character **R** on the SDSF panel) you will get an output like that shown in Figure 7-19 on page 100.

```

HZS1098I CHECK(IBMRA CF,RACF_GRS_RNL)
HZS1090I START TIME: 07/06/2006 14:00:54.161813
HZS1095I CHECK DATE: 20040703 CHECK SEVERITY: HIGH
IRRH250R
IRRH250R RACF_GRS_RNL Report
IRRH250R
IRRH251R S Major Minor Type QName Rname Typ
IRRH252R - - - - -
IRRH253R SYSZRACF SETROPTS SERNL
IRRH253R SYSZRACF DSDTDSDTDSDTDSDTDSDT SERNL
IRRH253R SYSZRACF DSDTPREPDSDTPREPDSDT SERNL
IRRH253R SYSZRACF RACF SERNL
IRRH253R SYSZRACF DSDTDSDTDSDTDSDTDSDT SERNL
IRRH253R SYSZRAC2 IRRCRV05 SERNL
IRRH253R SYSZRAC2 GLOBALGLOBALGLOBAL SERNL
IRRH253R SYSZRAC2 TEMPLATE-LOCK SERNL
IRRH253R SYSZRAC2 PROGRAMPROGRAMPROGRA SERNL
IRRH253R SYSZRAC2 DPDTABPT0000 SERNL
IRRH253R SYSZRAC2 ICHSEC00 SERNL
IRRH253R SYSZRAC2 IRRCRV05 SERNL
IRRH253R SYSZRAC2 IRRDPI080000 SERNL
IRRH253R SYSZRAC2 RCVTDPTB0000 SERNL
IRRH253R SYSZRAC2 XMCAXMCAXMCAXMCAXMCA SERNL
...
...
IRRH253R SYSZRACF CNSTGNLP*VMXEVENT SERNL
IRRH253R SYSZRAC2 RACGLIST_VTAMAPPL SERNL
IRRH253R SYSZRACF CNSTRCLP*VTAMAPPL SERNL
IRRH253R SYSZRACF CNSTGNLP*VTAMAPPL SERNL
IRRH253R SYSZRAC2 RACGLIST_VXMBR SERNL
IRRH253R SYSZRACF CNSTRCLP*VXMBR SERNL
IRRH253R SYSZRACF CNSTGNLP*VXMBR SERNL
IRRH253R SYSZRAC2 RACGLIST_WIMS SERNL
IRRH253R SYSZRACF CNSTRCLP*WIMS SERNL
IRRH253R SYSZRACF CNSTGNLP*WIMS SERNL
IRRH253R SYSZRAC2 RACGLIST_WRITER SERNL
IRRH253R SYSZRACF CNSTRCLP*WRITER SERNL
IRRH253R SYSZRACF CNSTGNLP*WRITER SERNL
IRRH253R SYSZRAC2 RACGLIST_XFACILIT SERNL
IRRH253R SYSZRACF CNSTRCLP*XFACILIT SERNL
IRRH253R SYSZRACF CNSTGNLP*XFACILIT SERNL

IRRH203I IRRH203I No RACF ENQ names were found in the GRS Resource Name List.

HZS1091I END TIME: 07/06/2006 14:00:54.179647 STATUS: SUCCESSFUL

```

Figure 7-19 Output of a successful check for RACF\_GRS\_RNL in debug mode

**Note:** Running a check in DEBUG mode can be useful when debugging a problem with the check.



For service aid and diagnosis purposes you can display the check in a diagnosis mode with the following command:

```
HZSPROC,DISPLAY,CHECKS,CHECK=(IBMRACF,RACF_IBMUSER_REVOKED),DIAG
```

The output in Figure 7-20 shows, in addition to the normal display, the address at which the check routine (here: IRRHCR00) is loaded.

```
HZS0201I 10.39.10 CHECK DETAIL      351
CHECK(IBMRACF,RACF_IBMUSER_REVOKED)
STATE: ACTIVE(ENABLED)              STATUS: EXCEPTION-MED
EXITRTN: IRRHCA00
LAST RAN: 07/06/2006 10:45    NEXT SCHEDULED: 07/07/2006 10:45
INTERVAL: 24:00
EXCEPTION INTERVAL: SYSTEM
SEVERITY: MEDIUM
WTOTYPE: EVENTUAL ACTION
SYSTEM DESC CODE: 3
THERE ARE NO PARAMETERS FOR THIS CHECK
REASON FOR CHECK:  IBMUSER should be revoked.
MODIFIED BY: N/A
DEFAULT DATE: 20051111
ORIGIN: HZSADDCK
LOCALE: HZSPROC
DEBUG MODE: OFF  VERBOSE MODE: NO
INTERNAL DIAGNOSTICS - CHECK TOKEN: 0102006A.7FC78000
ROUTINE: IRRHCR00-7F080FD0 MSGTBL: IRRHCM00-7F07E3D0  FUNC: CLEANUP
LAST CPU TIME: 0.319  MAX CPU TIME: 0.319
```

Figure 7-20 Check output of DISPLAY command with DIAG parameter

### Check report with exceptions

If this check finds an exception the output will look like that shown in Figure 7-21 and Figure 7-22 on page 102.

### RACF\_GRS\_RNL check report with exceptions:

START TIME: 11/10/2004 10:13:10.341622 IBMRACF, RACF\_GRS\_RNL  
OWNER DATE: 20040703

#### RACF\_GRS\_RNL Report

S	Major	Minor	Type	QName	Rname	Type
E	SYSZRACF	SETROPTS	SERNL	SYSZRACF	SETROPTS	SPEC
E	SYSZRAC2	IRRCRV05	SERNL	SYSZRAC2	IRRCRV05	SPEC
E	SYSZRAC2	IRRCRV05	SIRNL	SYSZRAC2	IRRCRV05	SPEC
E	SYSZRAC5	ALIAS	SERNL	SYSZRAC5	AL	GEN

\* High severity Exception \*

IRRH202E One or more RACF ENQ names were found in a GRS Resource Name List.

#### Explanation:

The RACF RACF\_GRS\_RNL check has detected that a RACF resource is covered by an entry in the specified GRS resource name list (RNL). RACF resource names should not be in either the system inclusion RNL (SIRNL) or the system exclusion RNL (SERNL).

#### System Action:

The check continues processing. There is no effect on the system.

#### Operator Response:

Report this problem to the system programmer.

#### System Programmer Response:

Ensure that the RACF resource names are removed from the specified resource name list (RNL).

Figure 7-21 Output with an exception for RACF\_GRS\_RNL

#### Problem Determination:

See "MVS Planning: Global Resource Serialization" for details on resource name lists (RNLs). Ensure that the RACF ENQ names do not match any of your resource name list entries. A list of the RACF ENQ names may be found in the RACF Systems Programmer's Guide.

#### Source:

RACF Systems Programmer's Guide

#### Reference documentation:

RACF Systems Programmer's Guide MVS Planning: Global Resource Serialization

#### Automation:

None.

IBMRACF Reason: None of the RACF ENQ names should be in RNLs.

Check parameters: N/A

END TIME: 01/08/2005 20:47:54.819710 RESULT: 0000000C DIAG:  
00000000\_00000000

Figure 7-22 Output with an exception for RACF\_GRS\_RNL (continued)

## 7.4.2 Check RACF\_SENSITIVE\_RESOURCES

This check examines the security characteristics of several system-critical data sets (for example, APF data sets) and general resources.

The check verifies the protection of each resource by extracting its profile and examining the UACC, WARNING status, and the ID(\*) entry in the access list if one exists. In addition, if there is no profile protecting a data set, and if NOPROTECTALL or PROTECTALL(WARN) is in effect, the check flags the data set as an exception.

The customer can optionally specify a user ID to the check which, if specified, is used to perform a RACF authorization check for the next higher access authority after the highest expected general access authority.

**Note:** IBM recommends that you protect system-critical resources properly to avoid a potential system exposure.

Table 7-7 Main criteria for the RACF\_SENSITIVE\_RESOURCES

Criteria	Value
OWNER	IBMRACF
Severity	high
WTotype	critical
Interval	04:00
Reason	Sensitive resources should be protected.
Debug mode	off

Compared to the RACF checks that were previously described, it is noticeable that this check runs more frequently (every 4 hours). Like the RACF\_GRS\_RNL check, it has a high severity.

### Check output

Until z/OS V1R7 the check output of this check consisted of the following sub-reports:

- ▶ APF Dataset Report
- ▶ RACF Dataset Report

Since z/OS V1R8 the following reports were added:

- ▶ PARMLIB Dataset Report  
The parmlib data sets names are extracted using IEFPRMLIB.
- ▶ Current Link List Dataset Report  
The link list data set names are extracted using CSVDYNL.
- ▶ Sensitive General Resources Report  
The general resource list is a hardcoded list of class/resource name pairs. Further processing is handled by routines modelled after the existing data set list (processGeneralResouce and getRacGrInfo).

The output of this check is a list of exceptions flagged. For each of these, the check examines:

- ▶ For system-critical data sets, that the data set exists on the expected volume. If the data set does not exist on the volume, a V (volume exception) is placed in the Status (S) column.
- ▶ That the resource has baseline protection. For example, APF data sets can have a general access as high as READ, while the data sets which comprise the RACF database

must have a general access of NONE. An E in this column indicates that the check found an exception and that there is excessive access authority allowed to the data set.

### Successful check

If the check did not find any exception, there will be no flags (E or V) in the status column, and the following message confirms that there is no deviation from the recommendation:

IRRH205I The RACF check RACF\_SENSITIVE\_RESOURCES has not found any errors in the security controls on this system.

### Check report with exceptions

If the check finds exceptions you will find following sub-reports:

The AFP™ Dataset Report in Figure 7-23 on page 105 shows a lot of exceptions. For example, the data set SYS1.VTAMLIB has a UACC of ALTER and the TCPIP data sets are not protected at all. There is no profile for it and the system wide option PROTECT-ALL is not activated.

```
CHECK(IBMRACF,RACF_SENSITIVE_RESOURCES)
START TIME: 07/07/2006 06:46:05.049680
CHECK DATE: 20040703  CHECK SEVERITY: HIGH
```

#### APF Dataset Report

S Data Set Name	Vol	UACC	Warn	ID*	User
-----					
E \$DSN810.SDSNLINK	SBOX11				
V ADB.V5R1M0.SADBLINK	SBOX91				
V APK.ACIF.SAPKMOD1	Z18RB1				
E ARS.SARSLMD5	SBOXC7				
E ARS.SARSLOAD	SBOXC7				
E ARSUSER.LOADLIB	SBOXC7				
V ASN.V8R1M0.SASNLOAD	SBOX91				
V AUO.V2R1M0.SAUOLOAD	SBOX91				
E BB05S63.SBBOLD2	SBOX36				
E BB05S63.SBBOLoad	SBOX36				
E BB05S63.SBBOLPA	SBOX36				
...					
...					
E SYS1.SISTCLIB	Z18RB1	Altr No		****	
E SYS1.SVCLIB	Z18RB1	Altr No		****	
V SYS1.SYSPROG.CSSLIB	SBOX01				
V SYS1.SYSPROG.LINKLIB	SBOX01				
V SYS1.SYSPROG.LPALIB	SBOX01				
V SYS1.SYSPROG.MIGLIB	SBOX01				
E SYS1.USER.LOAD	BH5CAT				
E SYS1.VTAMLIB	Z18RB1	Altr No		****	
E TCPIP.SEZADSIL	Z18RB1				
V TCPIP.SEZALINK	Z18RB1				
E TCPIP.SEZALNK2	Z18RB1				
E TCPIP.SEZALOAD	Z18RB1				
E TCPIP.SEZALPA	Z18RB1				
V TCPIP.SEZAMIG	Z18RB1				
E TCPIP.SEZATCP	Z18RB1				
...					
...					
V WTSCPLX2.EJES41J2.LINKLIB	SBOXEC				
E WTSCPLX2.EJES41J3.LINKLIB	SBOXB4				

Figure 7-23 Output of RACF\_SENSITIVE\_RESOURCES: APF Dataset Report

The **RACF Dataset Report** in Figure 7-24 on page 106 lists the exceptions for the active RACF databases. They are protected by a profile that is defined with UACC(ALTER). This does not correspond to IBM recommendations.

RACF Dataset Report					
S	Data Set Name	Vol	UACC	Warn	ID* User
E	SYS1.RACFESA	BH5CAT	Altr	No	****
E	SYS1.RACFBK	BH5CAT	Altr	No	****

Figure 7-24 Output of RACF\_SENSITIVE\_RESOURCES: RACF Dataset Report

The PARMLIB Dataset Report in Figure 7-25 shows the exceptions for the active parmlibs. The CPAC.PARMLIB has a UACC(UPDATE). The other two parmlibs have an exception for which you do not know the reason exactly. You have to do further investigation to find the reason for the exceptions. It might come from a universal access (UACC) or ID(\*) access list entry which is too permissive, or if there is no profile, then PROTECTALL(FAIL) is not in effect.

You can determine if there is a fitting profile by specifying the following LISTDSD command:

```
LD DA('SYS1.PARMLIB') AUTH GEN
```

You can further look at the system-wide option like PROTECTALL(FAIL) using:

```
SETROPTS LIST
```

**Note:** The parmlibs are listed in alphabetic order, not in the order of the valid concatenation.

PARMLIB Dataset Report					
S	Data Set Name	Vol	UACC	Warn	ID* User
E	CPAC.PARMLIB		Updt	No	****
E	SYS1.IBM.PARMLIB				
E	SYS1.PARMLIB				

Figure 7-25 Output of RACF\_SENSITIVE\_RESOURCES: PAMRLIB Dataset Report

Current Link List Dataset Report					
S	Data Set Name	Vol	UACC	Warn	ID* User
-	-----	----	----	----	----
E	ASM.SASMMOD1	Z18RB1			
E	ASM.SASMMOD2	Z18RB1			
E	CBC.SCCNCMP	Z18RB1			
E	CBC.SCLBDLL	Z18RB1			
E	CBC.SCLBDLL2	Z18RB1			
E	CEE.SCEERUN	Z18RB1			
E	CEE.SCEERUN2	Z18RB1			
E	CSF.SCSFMODE	Z18RB1			
E	EJES.SEJEHENU	Z18RB1			
E	EJES.SEJELINK	Z18RB1			
E	EOX.SEPHLOD1	Z18RB1			
E	EOY.SEOYLOAD	Z18RB1			
E	EUV.SEUVLINK	Z18RB1			
E	FFST.V120ESA.SEPWMOD1	Z18RB1			
E	FFST.V120ESA.SEPWMOD2	Z18RB1			
E	FFST.V120ESA.SEPWMOD4	Z18RB1			
E	PLEX75.DB2V8.SDSNLINK	BH5ST2			
E	SYS1.CMDLIB	Z18RB1	Altr	No	****
E	SYS1.CSSLIB	Z18RB1	Altr	No	****
E	SYS1.DGTLLIB	Z18RB1	Altr	No	****
...					
...					

Figure 7-26 Output of RACF\_SENSITIVE\_RESOURCES: Current Link List Dataset Report

Sensitive General Resources Report					
S	Resource Name	Class	UACC	Warn	ID* User
-	-----	----	----	----	----
	BPX.DAEMON	FACILITY	None	No	****
	BPX.FILEATTR.APF	FACILITY	None	No	****
E	BPX.SERVER	FACILITY	Altr	No	****
	BPX.SUPERUSER	FACILITY	None	No	****
E	ICHBLP	FACILITY	Altr	No	****
E	IRR.PASSWORD.RESET	FACILITY	Altr	No	****
E	MVS.SET.PROG	OPERCMDS	Ctrl	No	****
E	MVS.SETPROG	OPERCMDS	Ctrl	No	****
	ACCT	TSOAUTH	None	No	****
	CONSOLE	TSOAUTH	None	No	****
	OPER	TSOAUTH	None	No	****
	PARMLIB	TSOAUTH	None	No	****
	TESTAUTH	TSOAUTH	None	No	****
	SUPERUSER.FILESYS	UNIXPRIV			
	SUPERUSER.FILESYS.CHANGEPARMS	UNIXPRIV			
	SUPERUSER.FILESYS.CHOWN	UNIXPRIV			

Figure 7-27 Output of RACF\_SENSITIVE\_RESOURCES: Sensitive General Resources Report

**\* High Severity Exception \***

IRRH204E The RACF\_SENSITIVE\_RESOURCES check has found one or more potential errors in the security controls on this system.

Explanation: The RACF security configuration check has found one or more potential errors with the system protection mechanisms.

System Action: The check continues processing. There is no effect on the system.

Operator Response: Report this problem to the system security administrator and the system auditor.

System Programmer Response: Examine the report that was produced by the RACF check. Any data set which has an "E" in the "S" (Status) column has excessive authority allowed to the data set. That authority may come from a universal access (UACC) or ID(\*) access list entry which is too permissive, or if the profile is in WARNING mode. If there is no profile, then PROTECTALL(FAIL) is not in effect. Any data set which has a "V" in the "S" (Status) field is not on the indicated volume. Remove these data sets from the list or allocate the data sets on the volume. Any data set which has an "M" in the "S" (Status) field has been migrated.

The APF\_LIBS check provides additional analysis of the non-RACF aspects of your APF list.

If the "S" field contains an "E" or is blank, then blanks in the UACC, WARN, and ID(\*) columns indicate that there is no RACF profile protecting the data set. Data sets which do not have a RACF profile are flagged as exceptions, unless SETROPTS PROTECTALL(FAIL) is in effect for the system.

If a valid user ID was specified as a parameter to the check, that user's authority to the data set is checked. If the user has an excessive authority to the data set, that is indicated in the USER column. For example, if the user has ALTER authority to an APF-authorized data set, the USER column contains ">Read" to indicate that the user has more than READ authority to the data set.

Problem Determination: See the RACF System Programmer's Guide and the RACF Auditor's Guide for information on the proper controls for your system.

**Source:**

RACF System Programmer's Guide  
RACF Auditor's Guide

**Reference Documentation:**

RACF System Programmer's Guide  
RACF Auditor's Guide

Automation: None.

Check Reason: Sensitive resources should be protected.

END TIME: 07/07/2006 06:46:08.488144 STATUS: EXCEPTION-HIGH

*Figure 7-28 Output of RACF\_SENSITIVE\_RESOURCES: Explanation*

At the end of the report you find the overall explanation for the exception, as shown in Figure 7-28 on page 108.



## LDAP change logging

This chapter demonstrates how z/OS V1R8 provides a solution to some functional gaps in the way that change logging of RACF profile updates were reflected in z/OS LDAP. In addition it describes an enhancement to the LISTUSER command with regard to password enveloping enabled for a user.

The enhancements covered are the following:

- ▶ GROUP updates can now be change logged with:
  - NOTIFY.LDAP.GROUP
- ▶ CONNECT updates can now be change logged with:
  - NOTIFY.LDAP.CONNECT
- ▶ A generic profile can now be used to change log, USER, GROUP and CONNECT
  - NOTIFY.LDAP.\*
- ▶ Updates to the LISTUSER command output show the existence of password envelopes.
- ▶ Change log entries are now created for all new passwords even when not enveloped.

## 8.1 LDAP overview

The z/OS Lightweight Directory Access Protocol (LDAP) server is a part of the Integrated Security Services for z/OS. It is based on a client/server model that provides client access to an LDAP server. LDAP is an open industry standard that defines a standard method for accessing and updating information in a directory.

LDAP defines a communication protocol. That is, it defines the transport and format of messages used by a client to access data running over TCPIP. LDAP does not define the directory service itself.

### RACF and LDAP processing

As shown in Figure 8-1 on page 111, LDAP communicates with RACF through SDBM, the RACF database back end of the LDAP server. The SDBM database allows for directory authentication (or bind) using the RACF user ID and password. The RACF user ID must have an OMVS segment defined and an OMVS UID present. The RACF user and group information that make up an identity can be used to establish access control on other LDAP directory entities. This expands the use of the RACF identity to the rest of the LDAP-managed namespace.

The LDAP server is a z/OS UNIX application, with run-time parameters provided in the configuration file called slapd.conf. The configuration file allows you to enable the back ends.

The SDBM back end externalizes the USER and GROUP profiles in RACF as though they were entries in a directory tree. The authenticated LDAP user can use LDAP operations to maintain these profiles provided that they have the privilege to do so in RACF. Using the SDBM back end of the LDAP server you can:

- ▶ Add new users and groups to RACF
- ▶ Add users to groups (connections)
- ▶ Modify RACF information for users and groups
- ▶ Retrieve RACF information for users and groups
- ▶ Delete users and groups from RACF
- ▶ Remove users from groups (connections)

The **TDBM** back end uses DB2 tables to store any kind of data that users wish to manipulate. It is used by the GDBM that uses the DB2 tables to store its data.

The **GDBM** (change log back end) maintains a directory of Change Log entries; each entry contains information about a single change and can be inspected by an LDAP client. Note that the GDBM naming space suffix is fixed to “cn=changelog”, as shown in Figure 8-2 on page 111.

**Reminder:** LDAP uses the distinguished name (DN) syntax to identify any object in a directory naming space. This pertains to users as well; when identification is required, the user is eventually designated in the LDAP processes with his distinguished name.



## 8.2 Change log processing prior to z/OS V1R8

Prior to z/OS V1R8, RACF required that the profile be defined. The profile acts as a switch to turn on or off RACF event notification.

**Note:** To have RACF notify the LDAP server to log a change, a new profile in the RACFEVNT class was defined, NOTIFY.LDAP.USER, as shown in Figure 8-3.

```
RDEF RACFEVNT NOTIFY.LDAP.USER UACC(NONE)
SETR RACLIST(RACFEVNT) REFRESH
```

Figure 8-3 RACF commands to activate event notification

Once the switch is turned on, RACF changes are reflected in the GDBM back-end directory. For example, the changes to a RACF USER profile that results in creating a change log entry in the GDBM back end are the following:

- ▶ Password changes, regardless of the interface used, as long as the new password had to be enveloped
- ▶ Updates to a user's revoke status
- ▶ User additions made using the ADDUSER command
- ▶ User modifications made using ALTUSER and PASSWORD commands
- ▶ User deletions made using the DELUSER command

## 8.3 Change log processing enhancements in z/OS V1R8

The change log is a set of entries in the directory that contain information about changes to objects. Depending on configuration options, information about a change to a TDBM entry or to an object controlled by an application (for example, a RACF user, group, or user-group connection profile) can be saved in a change log entry. An LDAP search operation can be used to retrieve change log entries to obtain information about what changes have taken place.

Currently, z/OS LDAP supports the query and update of USER, GROUP, and group connection attributes using the SDBM back end to talk to RACF. RACF currently supports LDAP change logging of updates to USER profiles. Thus, there is a functionality missing in RACF change logging with respect to the RACF functions supported by z/OS LDAP. z/OS V1R8 includes a support change for logging of group and connection updates.

### Group change logging

Group change logging now allows an LDAP client to be notified of a RACF-initiated change for any of the profile types supported by the z/OS LDAP server. RACF can now be configured to create LDAP change log entries in response to changes in user and group profiles. In addition, all password changes are logged whether they are enveloped or not.

This provides an open, remote method of change notification. An LDAP client can read the LDAP change log, detect updates to RACF users, groups, and group membership, and then retrieve RACF entries using only LDAP interfaces. To use this function, the LDAP server must be configured to enable the SDBM back end.

## Event notifications

Event notifications, through the creation of LDAP change log entries, are controlled by RACF resources in the RACFEVNT class. Two new resource profiles have been created in this class for group and connect to allow LDAP change log entries to be created for the corresponding event types on a system-wide basis.

In addition, a new line in the LISTUSER command output has been added to demonstrate the existence of password envelopes, and there is now unconditional change logging of all password updates. This enhancement solves the problems from previous releases that there was no indication in the LISTUSER command as to existence of a password envelope and no change log entry was created for a new password which was not enveloped.

## 8.4 Activating LDAP change notification

RACF can be configured to create LDAP change log entries in response to changes in user and group profiles. This provides an open, remote method of change notification. An LDAP client can read the LDAP change log, detect updates to RACF users, groups, and group membership, and then retrieve RACF entries using only LDAP interfaces. To use this function, the LDAP server must be configured to enable the SDBM back end.

Event notifications, through the creation of LDAP change log entries, are controlled by RACF resources in the RACFEVNT class. If the RACFEVNT class is active, and the appropriate resource is protected by either a discrete or generic profile, LDAP change log entries are created for the corresponding event types on a system-wide basis.

To activate LDAP change notification in RACF, you need to define, in the RACFEVNT class, resources for the notifications you want to log, and then activate the RACFEVNT class. Two new resources NOTIFY.LDAP.GROUP and NOTIFY.LDAP.CONNECT are introduced in the RACFEVNT class.

**Note:** You do not need to define resources in the LDAPBIND class to enable LDAP change notifications.

### Defining RACFEVNT profiles

Define the RACFEVNT resource class profiles for the LDAP notifications you want by creating one or more discrete profiles or by creating a generic profile, and activating them with a SETROPTS command, as follows:

```
RDEFINE RACFEVNT NOTIFY.LDAP.USER
RDEFINE RACFEVNT NOTIFY.LDAP.GROUP      - New with z/OS V1R8
RDEFINE RACFEVNT NOTIFY.LDAP.CONNECT    - New with z/OS V1R8
```

```
SETROPTS GENERIC(RACFEVNT)
RDEFINE RACFEVNT NOTIFY.LDAP.*
```

```
SETROPTS CLASSACT(RACFEVNT) RACLIST(RACFEVNT)
```

If a resource is defined in the RACFEVNT class called NOTIFY.LDAP.USER, an LDAP change log entry is created when a user's password is changed.

Using NOTIFY.LDAP.GROUP in the RACFEVNT class results in change log entries for:

- ▶ Additions made using the ADDGROUP command
- ▶ Modifications made using the ALTGROUP command

- Deletions made using the DELGROUP command

Figure 8-2 on page 111 shows a group change log entry.

Using NOTIFY.LDAP.CONNECT in the RACFEVNT results in change log entries for:

- Additions and modifications made using the CONNECT command
- Deletions using the REMOVE command
- Establishment of the connection of a user to its default group by the ADDUSER command
- Modification to a user's connection information using the GROUP, UACC, and AUTHORITY operands of the ALTUSER command

Figure 8-4 shows a connect change log entry.

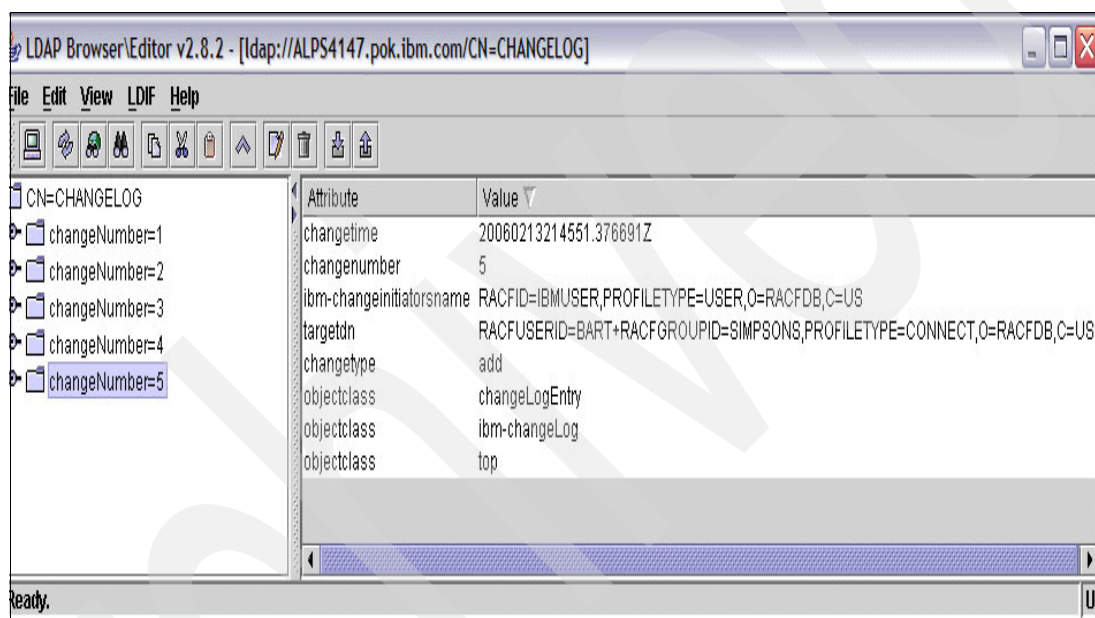


Figure 8-4 Connect change LDAP change entry log

**Note:** For the LDAP distinguished name format of the RACF CONNECT profile name in the targetdn attribute, LDAP performs this name mapping.

## 8.5 Password enveloping enhancements

Password envelopes enable authorized applications to recover user passwords. Password envelopes are used by IBM Directory Integrator, in conjunction with LDAP notification, to enable a heterogeneous password synchronization solution.

RACF can be configured to save user passwords such that the clear text can be recovered by an authorized application. This ability can be restricted to a subset of your users. When an eligible user's password is changed, the new password is encrypted under a public key contained within a key ring associated with the RACF subsystem address space identity. The encrypted password is then stored in the user's profile. When an application requests the password, RACF decrypts the password, and then encrypts it in PKCS #7 format for recipients whose digital certificates have been placed on the same RACF key ring. The application can then decrypt the password envelope using its private key.

## PASSWORD.ENVELOPE profile

The PASSWORD.ENVELOPE profile in the RACFEVNT class controls whether new passwords are enveloped for a given user. If the user whose password is being changed has at least READ access to this resource, then the new password is enveloped, as follows:

```
PERMIT PASSWORD.ENVELOPE CLASS(RACFEVNT) ID(USER1 USER2 GROUPA GROUPB)
ACCESS(READ)
```

Detailed descriptions of the usage and invocation of password envelopes can be found in the *z/OS Security Server RACF Security Administrators Guide*, SA22-7683.

**Note:** The LISTUSER command indicates the presence of a password envelope when RACFEVNT class is active and a PASSWORD.ENVELOPE profile exists.

## IRRDBU00 utility

The database unload utility, IRRDBU00, indicates the presence of password envelopes; however, it is less convenient than a LISTUSER and an administrator with authority to issue a LISTUSER command may not be authorized to run the IRRDBU00 utility.

A LISTUSER command indicates the presence of a password envelope if either of the two following condition are met:

- ▶ The RACFEVNT class is active and a PASSWORD.ENVELOPE profile exists.
- ▶ The user has a residual envelope.

**Note:** Changes are logged to SMF records for each privileged operation associated with password enveloping, such as changes to the key ring, changes to the enveloping policy, and actual retrievals of password envelopes from RACF.

The new **LISTUSER** command output line (shown in bold in Figure 8-5) indicates whether a password is enveloped.

```
USER=ACE  NAME=UNKNOWN  OWNER=WELLIE
CREATED=92.162
DEFAULT-GROUP=KINGS  PASSDATE=00.000  PASS-          INTERVAL=N/A
PHRASEDATE=N/A
PASSWORD ENVELOPED=NO
ATTRIBUTES=NONE
REVOKE DATE=NONE  RESUME DATE=NONE
LAST-ACCESS=06.044/12:26:08
CLASS AUTHORIZATIONS=NONE
NO-INSTALLATION-DATA
NO-MODEL-NAME
```

Figure 8-5 Example of a LISTUSER command showing the PASSWORD ENVELOPE field

## z/OS V1R8 enhancements

The enveloped password is not displayed in the user's LISTUSER output. (The line PASSWORD ENVELOPED=YES in the LISTUSER output indicates that a password envelope is present.

The enveloped password is not unloaded by the database unload (IRRDBU00) utility. (A field in the output indicates that a password envelope is present.

If the user fails verification (when the RACROUTE REQUEST=VERIFY is executed during password envelope processing), the user's new password is not enveloped, even when the password change is successful. One possible reason for the verification failure (during password envelope processing) is that the user is revoked at the time that password envelope processing occurs.

For example, if an administrator uses the ALTUSER command to change the password of a revoked user who is eligible for password enveloping, the user's password is changed but the user's password is not enveloped. Even when the administrator subsequently resumes the revoked user, the password is not enveloped.

To envelope the password of an eligible user who is revoked, you must resume the user before the password change, or resume the user with the same ALTUSER command that changes the password. For example:

```
ALTUSER userid PASSWORD(new-password) RESUME
```

When you use this example, the revoked user's new password is enveloped.

## 8.6 Change logging of password changes

As of z/OS V1R8, a password change will always result in a change log entry if the RACF NOTIFY.LDAP.USER profile is defined in the RACFEVNT class. This no longer depends on whether the password was enveloped, but applies equally to all password changes and where there are new potential values in the changes attribute of a user-related LDAP change log entry. A well-written LDAP client, such as the IBM Tivoli Directory Integrator, should not be affected by these changes.

When the password is enveloped, the existing behavior continues, as follows:

- ▶ The \*ComAndGetIt\* string in the changes attribute of the change log entry is created.

When the password is not enveloped:

- ▶ A new \*NoEnvelope\* string in the changes attribute is added to the change log entry, as shown in Figure 8-6 on page 117.



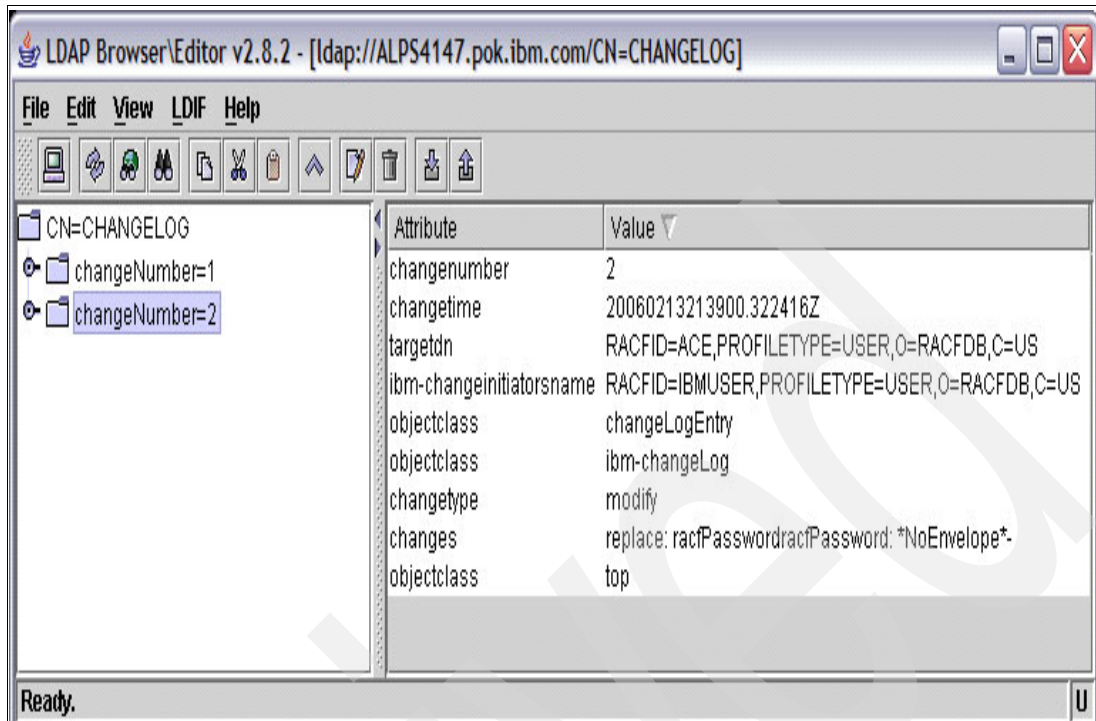


Figure 8-6 Example of password change log entry

**Note:** A password may not be enveloped because:

- The user is not eligible for enveloping.
- The password contains invalid characters.
- The enveloping option failed due to errors.

## Changes to messages

RACF messages IRRRC131I and IRRRC135I have been altered so that the *class name* field can now be USER, GROUP or CONNECT, as follows:

**IRRC131I** RACF ENCOUNTERED AN R\_PROXYSERV ERROR WHILE ATTEMPTING TO CREATE AN LDAP CHANGE LOG ENTRY FOR AN UPDATE TO *class name*. SAF RETURN CODE=SAF-*return-code*, RACF RETURN CODE=return-code, RACF REASON CODE=*reason-code*.

**IRRC135I** RACF ENCOUNTERED AN EXTRACT ERROR FOR PROFILE profile-name IN CLASS *classname* WHILE PROCESSING *classname2 name*. RETURN CODE=return-code, RACF RETURN CODE=*racf-return-code*, RACF REASON CODE=*racf-reason-code*.

The following message explanations have been altered to accommodate group/connect change logging.

**IRRC132I** RACF ENCOUNTERED AN UNEXPECTED PARSE RETURN CODE nn WHILE PROCESSING AN IRRLOGOO COMMAND.

**IRR417I** UNABLE TO COMMUNICATE WITH THE RACF SUBSYSTEM. IEFSSREQ RETURN CODE IS return-code.

Archived

## Template and profile extensions

This chapter discusses the RACF template extensions that have been introduced in z/OS V1R8. The RACF database is formatted into 4K blocks and RACF can only handle a template that fits into a 4K block. The problem is that the USER block is nearly full. The solution to this problem is change RACF initialization and utility processing to allow the templates to expand beyond 4K. As a result, special consideration must be given to sharing a z/OS V1R8 RACF database with a previous release level of z/OS. Applications that read the RACF database directly will be affected by this change as well.

**Note:** This support is being rolled back to z/OS V1R4. V1R5, V1R6, and V1R7 with APAR OA12443 to allow earlier releases to share a RACF database that might have expanded templates.

In addition, also described are some extensions to RACF profiles that provide:

- ▶ New class attributes for disallowing generic profiles in a class
- ▶ IRRDPI00 list command granularity
- ▶ KERBLINK class enhancements to accept mixed-case profile names
- ▶ Changes to the ADDUSER and ALTUSER OMVS FILEPROC MAX keyword limit

## 9.1 RACF database template extensions

The RACF database contains records whose format is controlled by a set of database templates. Each RACF profile type is described in a separate template. The templates map out how profiles are written on the RACF database, providing the attributes of each field within the profile type. Each profile is described in a different template:

- ▶ GROUP
- ▶ USER
- ▶ CONNECT
- ▶ DATASET
- ▶ GENERAL RESOURCE

IBM makes changes to the templates to add new segments to the RACF database, or to add new fields to existing segments. The templates are shipped in a CSECT named IRRTEMP2. The RACF initialization IRRMIN00 utility can be used to format a new RACF database or to update an existing database with a new set of IBM-supplied RACF templates.

There are three copies of the templates:

- ▶ The latest version shipped with RACF is in the CSECT IRRTEMP2.
- ▶ The RACF database contains a copy of the templates. This copy of the templates controls how programs that access the database directly, such as IRRUT200, process RACF database records.
- ▶ There is an in-storage copy, anchored in the RCVT, which most RACF commands and processes other than the utilities use. This copy controls how users and programs that access the database through RACF process the database records. RACF initialization builds the in-storage copy at IPL time.

Refer to the following manuals for a detailed description of RACF templates and the IRRMIN00 utility:

- ▶ *z/OS Security Server RACF Systems Programming Guide, SA22-7681*
- ▶ *z/OS Security Server RACF Macros and Interfaces, SA22-7682*
- ▶ *z/OS Security Server RACF RACROUTE Macro Reference, SA22-7692*

**Note:** Any time you install a new release of RACF, or install a PTF that includes new templates, you need to insure that all three copies of the templates are at the same level. Use the RACF utility IRRMIN00 to do this.

### 9.1.1 Applications that read the RACF database directly

The RACF database is formatted into 4K blocks. RACF can only handle a template that will fit into one 4K block and the USER template is almost full. In order to address this a change has been made to RACF initialization and utility processing to allow for any templates to expand beyond one 4K block. This expansion is transparent to applications that use the intended interfaces to process RACF database fields such as:

- ▶ RACROUTE
- ▶ ICHEINTY, ICHEACTN and ICHETEST
- ▶ RACF commands
- ▶ RACF callable services

**Important:** Any application that reads and processes the RACF database templates directly may be affected.

## RACF database template structure

Figure 9-1 illustrates the first ten blocks on the RACF database, blocks 0 through 9, as follows:

- ▶ Block 0 contains the header information or the inventory control block (ICB).
- ▶ Blocks 1 through 9 are allocated to hold RACF templates, where each template describes a profile type, for example GROUP or GENERAL RESOURCE.
- ▶ The ICB (inventory control block), while not a programming interface, is documented in the *z/OS Security Server RACF Diagnosis Guide*, GA22-7689. Field ICTMPXCT was added to hold the number of expansion blocks.

Prior to z/OS V1R8, RACF could only support a template whose fields all fit into a 4K block. Since at least one profile type is now approaching the 4K limit, z/OS V1R8 now supports a template that has grown beyond the old 4K limit.

Blocks									
0	1	2	3	4	5	6	7	8	9
ICB	T	E	M	P	L	A	T	E	S

Figure 9-1 RACF database template structure

## An expanded template

As shown in Figure 9-2 on page 122, block X and block XXX are within blocks 1 through 9 of the RACF database. If any application directly reads the RACF templates, it now needs to check the last three bytes in each template block to see if the template has expanded into another block. When a template expands to larger than 4K, it will be continued into another 4K block. The last three bytes now contain the RBA (relative block address) of the block that contains more fields or segments for the profile type of this template block.

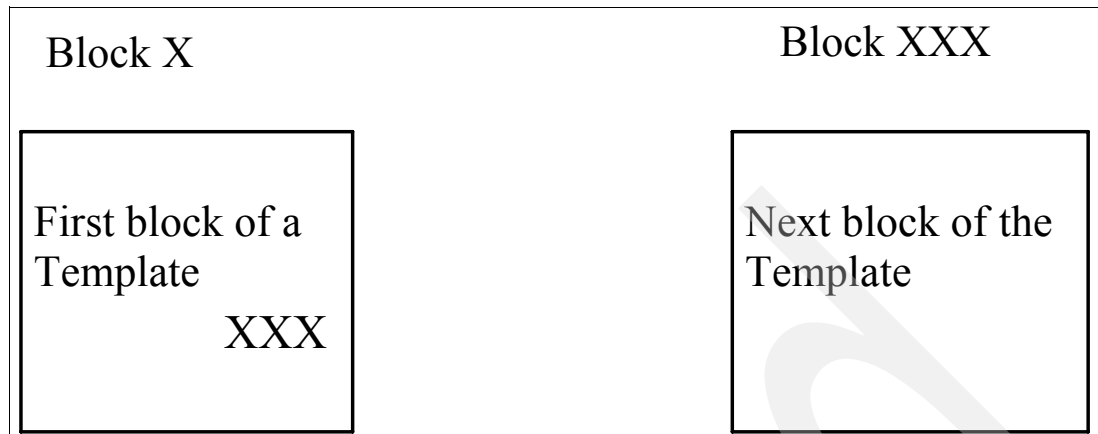


Figure 9-2 New expanded templates

### 9.1.2 Migration considerations

When migrating to z/OS V1R8, consider the following migration and coexistence considerations within a sysplex environment.

#### Sharing the V1R8 RACF database with a previous release level

Since RACF databases can be shared between RACF releases at different levels, the support for templates expanding to greater than 4K is rolled back to support all RACF releases, z/OS V1R4, V1R5, V1R6, and V1R7. Any system that shares a RACF database with a z/OS V1R8 should install a toleration APAR OA12443. The support in the APAR will allow an earlier level system to recognize when a RACF database template has expanded into a second 4K block and handle it appropriately.

**Note:** IBM does not support z/OS V1R4 and z/OS V1R8 coexisting. However, since a RACF database can be shared outside of a sysplex environment, there is support to share a RACF database between V1R4 and V1R8 with some restrictions.

If you share a V1R8 system with a lower level system without the APAR installed on the lower level release, errors may occur when RACF attempts to retrieve a database field which has its definition in the expanded template block. In addition RACF utilities may report on the RACF database incorrectly.

**Note:** You can also share a RACF database between z/OS V1R8 and z/VM®, no APAR is required for this. RACF databases on z/VM have not been kept current with z/OS.

#### Rules for the migration sharing a database

Always run IRRUT400 from highest level system. Run IRRMIN00 from highest level system or from the lower level system using JCL that includes a STEPLIB to an APF-authorized library that contains the z/OS V1R8 version of IRRMIN00.

With z/VM, always run utilities from the z/OS system.

Do not run the following utilities from a z/OS V1R4 system:

- ▶ IRRUT200
- ▶ IRRUT300 (BLKUPD)
- ▶ IRRDBU00

► IRRIRA00

Run these utilities from either z/OS V1R8 or from z/OS V1R5, z/OS V1R6, or z/OS V1R7 with APAR OA12443 installed.

## 9.2 Enhancements made to RACF profiles

Some IBM-defined classes never allow generic profiles, but prior to z/OS V1R8 there was no way to disallow generic profiles permanently in an installation-defined class. Generic processing is activated using the following commands:

SETROPTS GENERIC      Activates generic profile checking for a class

SETROPTS GENCMD      Activates generic command processing for a class

A new attribute, `GENERIC=ALLOWED | DISALLOWED`, specifies whether the SETROPTS GENERIC and SETROPTS GENCMD commands are ever allowed for the class. The default is `GENERIC=ALLOWED`.

Classes can be added to the CDT in either of the following ways:

- Dynamically by executing the RDEFINE CDTINFO command. They take effect immediately after a SETROPTS CLASSACT(MYCLAS8) is issued.
- Statically by updating the ICHERCDE macro followed by an assembly and linkedit. Static additions will only take place after an IPL.

This is described in detail in *z/OS Security Server RACF Systems Programming Guide*, SA22-7681.

### 9.2.1 Allowing or disallowing generics in static classes

An installation can add, modify, or delete installation-defined entries in the static class descriptor table using the ICHERCDE macro. There is a new keyword on the ICHERCDE macro:

- `GENERIC=ALLOWED | DISALLOWED`
  - `GENERIC=DISALLOWED` has been added to the following IBM-defined classes:
    - CDT
    - KERBLINK
    - REALM
    - SECLABEL
    - SECLMBR
- New ICHERCDE MNOTES have been added, as follows:
  - Generic values are invalid.
  - Member and `GENERIC=ALLOWED` are mutually exclusive.
  - `GENERIC=DISALLOWED` and `GENLIST=ALLOWED` are mutually exclusive.
  - Generic incompatible for shared POSIT value:nnnn.

### 9.2.2 Allowing or disallowing generics in dynamic classes

A dynamic class in RACF is defined with the RDEFINE with a CDTINFO keyword command, with `GENERIC(ALLOWED | DISALLOWED)` as a new keyword in the CDTINFO segment on

the RDEFINE and RALTER commands. The keyword specifies whether or not generic profile checking and generic command processing is allowed for a class. If not specified, the default for a class is GENERIC(ALLOWED).

An example is shown in Figure 9-3 on page 124.

```
//STEP1 EXEC PGM=IKJEFT01,DYNAMNBR=20
//SYSTSPRT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSTSIN DD *
RDEFINE CDT MYCLAS8 CDTINFO(POSIT(333) GENERIC(DISALLOWED) +
FIRST(ALPHA NUMERIC SPECIAL NATIONAL) +
OTHER(ALPHA NUMERIC SPECIAL NATIONAL))
```

*Figure 9-3 RDEFINE to disallow generic profiles in a class*

To activate a new class in the dynamic CDT, issue the following command:

```
SETROPTS CLASSACT(CDT) RACLIST(CDT)
```

If activating generics for a class that does not allow it, the following message is issued:

```
SETROPTS GENERIC(MYCLAS8)
ICH14075I SETROPTS GENERIC had no effect on class MYCLAS8.
```

### **RALTER command**

The RALTER command also has a new keyword GENERIC (ALLOWED | DISALLOWED). The NOGENERIC keyword changes the value to the default of GENERIC(ALLOWED).

### **RLIST command**

For the RLIST command, when the CDTINFO segment is displayed, a new line in the output will indicate the setting of the GENERIC keyword. The output from the RLIST command has been updated to indicate whether generics are allowed for the class, as shown in Figure 9-4 on page 125.



```

RLIST CDT MYCLAS8 CDTINFO
CLASS      NAME
-----
CDT        MYCLAS8

LEVEL  OWNER      UNIVERSAL ACCESS  YOUR ACCESS  WARNING
-----
  00    GGAINS          NONE              ALTER        NO

INSTALLATION DATA
-----
NONE

APPLICATION DATA
-----
NONE

AUDITING
-----
FAILURES(READ)
-----
NO USER TO BE NOTIFIED

CDTINFO INFORMATION
-----
CASE = UPPER
DEFAULTRC = 004
DEFAULTUACC = NONE
FIRST = ALPHA NATIONAL NUMERIC SPECIAL
GENLIST = DISALLOWED
GENERIC = DISALLOWED

```

Figure 9-4 RLIST command output showing generics disallowed

## New rules for generics in classes

The following rules must be followed when using generic profiles:

- ▶ Generic profiles are not allowed in a grouping class, so GENERIC(ALLOWED) cannot be specified with MEMBER for a dynamic class.  
For a static class, GENERIC=ALLOWED cannot be specified with the MEMBER= keyword.
- ▶ The GENLIST and GENERIC keywords must be compatible.
  - For a dynamic class, GENERIC(DISALLOWED) cannot be specified with GENLIST(ALLOWED).
  - For a static class, GENERIC=DISALLOWED cannot be specified with GENLIST=ALLOWED.
- ▶ Classes with shared posit numbers must have compatible settings of the GENERIC keyword. The exception is for grouping classes, which must specify GENERIC(DISALLOWED), and the corresponding member class, which can specify GENERIC(ALLOWED).

If the rule is violated and there is a mismatch between GENERIC keywords on classes with the same POSIT number, the following will apply:

- If the installation-defined class is sharing a POSIT number with an IBM class, then a GENERIC setting of the IBM class takes precedence and the installation class will be

changed to match the IBM class.

- If two installation classes, STATIC or DYNAMIC, share a POSIT number, the in storage copy will be changed to the least restrictive attribute GENERIC(ALLOWED), which takes precedence, and a class with GENERIC(DISALLOWED) is changed to GENERIC(ALLOWED).

**Note:** Even though the default on the ICHERCDE macro is GENERIC=ALLOWED, you can code MEMBER=class without specifying GENERIC=DISALLOWED. The default for a grouping class is still GENERIC=DISALLOWED. This was done for compatibility with previous releases.

### 9.2.3 New messages with z/OS V1R8

Following are some examples that produce new messages.

- Example 1 - Generic class attribute for a dynamic class.

```
RDEFINE CDT CLS1 CDTINFO (POSIT(333) GENERIC(DISALLOWED))
SETROPTS RACLIST(CDT) REFRESH
RDEFINE CDT CLS2 CDTINFO (POSIT(333) GENERIC(ALLOWED))
```

IRR52215I Warning: The attribute GENERIC(ALLOWED) in class CLS2 is not compatible with the attribute GENERIC(DISALLOWED) in class CLS1 because the classes share a POSIT number.

If you do not fix the problem, the following message is issued:

ICH14085I Warning: GENERIC(DISALLOWED) was changed to GENERIC(ALLOWED) for class CLS1 because the class shares a POSIT number with class CLS2.

- Example 2 - Static class and dynamic class definition in ICHRRCDE.

```
CLSX      ICHERCDE CLASS=CLSX,ID=1,GENERIC=DISALLOWED,POSIT=344
```

- After an IPL, CLSX is in the class descriptor table.
- Now define a dynamic class:

```
RDEFINE CDT CLSZ CDTINFO (POSIT(344) GENERIC(ALLOWED))
```

IRR52215I Warning: The attribute GENERIC(ALLOWED) in class CLSZ is not compatible with the attribute GENERIC=DISALLOWED in class CLSX because the classes share a POSIT number.

- If you do not fix this problem:

```
SETROPTS RACLIST(CDT) REFRESH
```

ICH14085I Warning: GENERIC=DISALLOWED was changed to GENERIC=ALLOWED for class CLSX because the class shares a POSIT number with class CLSZ.

#### Other new messages

New messages with RDEFINE and RALTER commands are the following:

IRR52212I Warning: GENERIC(ALLOWED) is ignored for profile class because MEMBER was also specified.

IRR52213I opt1 is not valid with opt2. You must correct this error before the class class can be added to the dynamic class descriptor table.

IRR52215I Warning: The attribute keyword-1 in class class-1 is not compatible with the attribute keyword-2 in class class-2 because the classes share a POSIT number.

A new message for the SETROPTS command or RACF initialization is:

ICH14085I Warning: keyword-1 was changed to keyword-2 for class class-1 because the class shares a POSIT number with class class-2.

## 9.2.4 Migration and coexistence considerations

If you share a RACF database between z/OS V1R8 and an earlier release you must follow these rules:

- ▶ Always administer a dynamic class that disallows GENERICS from a system running z/OS V1R8 or later.
- ▶ Always administer profiles where generics are disallowed from a system running z/OS V1R8 or later.

Otherwise you might inadvertently activate generic profiles processing for that class because generics cannot be disallowed on a back-level system.

### Panel changes for the generic keyword

The panels identified in Table 9-1 have been changed in support of the changes with the generic keyword.

Table 9-1 Panel changes for generic keyword support

Panel ID	Description
ICHP21N	Existing panel updated with CDTINFO GENERIC (ALLOWED   DISALLOWED) option
ICHH21N	Existing RDEF/RALT help panel update with GENERIC option
ICHH21N	New RDEF/RALT help panel for CDTINFO (GENERIC)
ICHM21	Existing message module updated with new messages
ICHS21N	Existing skeleton updated to build CDTINFO GENERIC

## 9.3 IRRDPI00 LIST command granularity

The IRRDPI00 LIST command lists every field definition in the RACF database, resulting in thousands of lines of output. The IRRDPI00 LIST command has new keywords that allow the specification of a single field, segment name, or profile type. By selecting a segment or a specific field, the output is much smaller and more readable.

```
IRRDPI00 LIST [(profile- type[segment-name [fieldname]])]
```

### Command examples

To list all fields in the OMVS segment of the USER profile, issue:

```
IRRDPI00 LIST(USER OMVS)
```

To list the HOME keyword in the OMVS segment of the USER profile, issue:

```
IRRDPI00 LIST(USER OMVS HOME)
```

```

PROFILE TYPE: USER
  <<<< ADD    SEGMENT DEFINITIONS >>>>
    SEGMENT NAME: OMVS
NOTIFY EXIT NAME: ICHCDXUP
  KEYWORD NAME: HOME
  TEMPLATE NAME: HOME
    VALIDITY CHECK NAME: ICHCDX12
    LIST PICTURE: CHARACTER
    LIST HEADING: /HOME=
    PROMPTING DATA: HOME DIRECTORY
    ASIS
    CHAR
    CHARACTER RESTRICTIONS: FIRST = ANY
                           OTHER = ANY

  PARAMETER TYPE: HOME
  HELP MESSAGES: OMVS USER'S INITIAL
                  WORKING DIRECTORY

```

### Command syntax errors

If you enter an incorrect profile, segment, or keyword combination, the following message is issued:

```
IRR52211I The IRRDPI00 LIST encountered an error. parameter not valid.
```

**Restriction:** To issue the IRRDPI00 command you need one of the following authorizations:

- ▶ RACF SPECIAL attribute
- ▶ Program access to IRRDPI00 module using RACF program control
- ▶ Read access to the IRRDPI00 resource in the FACILITY class

## 9.4 KERBLINK class enhancement

z/OS Security Server Network Authentication Service uses RACF to store and administer information about principles and realms. The general KERBLINK resource class maps principals to RACF user IDs on the system. Kerberos principle names can contain lower case letters. Prior to z/OS V1R8, profiles in the KERBLINK class could not be created with lower case letters because RACF did not support lower case characters when the KERBLINK class was introduced. Consequently some generated KERBLINK profile names containing lower case letters could not be manipulated.

The KERBLINK class has been updated to specify CASE=ASIS in the RACF class descriptor table to allow the KERBLINK class to accept mixed case profile names, so profiles can now be created that match a principle name with lower case characters.

### Mapping foreign principle names

All other foreign principals presenting tickets from the ITSO.IBM.COM server will be mapped to the ENDKERB user ID on the local z/OS system. Users HARRY and JANE have their foreign principal names mapped with individual user IDs on the local z/OS system, as follows:

```

RDEFINE KERBLINK /.../ITSO.IBM.COM/HARRY APPLDATA('PETRS')
RDEFINE KERBLINK /.../ITSO.IBM.COM/Jane APPLDATA('PAVELN')
RDEFINE KERBLINK /.../ITSO.IBM.COM/ APPLDATA('ENDKERB')

```

**Note:** The characters of the profile name are not translated to upper case, so be sure to enter the realm portion of the profile name in upper case and the foreign principal name in the appropriate case.

### 9.4.1 Migration considerations for KERBLINK class

If you have programs, scripts, or automated procedures that issue commands to manipulate profiles in the KERBLINK class, you must ensure that the KERBLINK profile names are in the proper case starting with z/OSV1R8.

Consider changing programs that specify the KERBLINK class profile names to use profile names in the proper case. This can be done by taking the following steps:

- ▶ Examine any programs, scripts, or automated procedures that issue RACF commands to manipulate profiles in the KERBLINK class.
- ▶ If those RACF commands contain lower case characters in the profile names, you must change them to upper case characters to produce equivalent commands on z/OS V1R8.
- ▶ Be sure to enter the realm portion of the profile name in upper case and the foreign principal name in the proper case.

#### Migration example

Consider a function like a REXX EXEC that issues the following RACF command (in lower case):

```
ralter kerblink /.../ITS0.ibm.com/alice apldata('asmith')
```

Change the command to the following:

```
ralter kerblink /.../ITS0.IBM.COM/ALICE apldata('asmith')
```

## 9.5 OMVS FILEPROCMAx change

Prior to z/OS V1R8, there was a limit on the maximum number of descriptors for files, sockets, directories, and any other file system object that a single process could concurrently have active or allocated.

For z/OS V1R7, the following limits were in effect:

- RACF - 524288
- USS - 131072

To get around this, increase the upper limits to 524287 for the following command:

```
ADDUSER or ALTUSER OMVS FILEPROCMAx
```

Now with z/OS V1R8, the new maximum limits are:

- RACF - 524287
- USS - 524287

**Note:** In RACF the maximum number of files a user is allowed to have concurrently active or allocated is reduced by one. In z/OS V1R8, UNIX System Services supports the larger value but only to 524287 so that one file descriptor is kept available for `exec()` processing. Therefore the RACF number was also reduced by one correspondingly.

## Use of the FILEPROC MAX keyword

You can limit the number of descriptors for files on a system-wide basis by updating the parmlib member BPXPRMxx MAXFILEPROC command. By limiting the number of open files that a process can have, you limit the amount of system resources a single process can use at one time. IBM Health Checker for z/OS can be used to determine whether the MAXFILEPROC value is set too low. Refer to 7.1, “IBM Health Checker for z/OS” on page 76 for more information.

You can set a system-wide limit in BPXPRMxx and then set higher limits for individual processes.

Use the RACF ADDUSER or ALTUSER command to specify the FILEPROC MAX limit on a per-process basis as follows:

```
ALTUSER userid OMVS(FILEPROC MAX(nnnn))
```

Refer to *z/OS UNIX System Services Planning*, SA22-7801 for a more detailed description.

## Migration and coexistence considerations

If you have already specified FILEMAXPROC(524288), UNIX System Services will use 524287 instead, with the following results:

- ▶ No change is required on the part of the security administrator.
- ▶ Any new **ALTUSER** commands will not accept 524288.

# Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this redbook.

## IBM Redbooks

For information on ordering these publications, see “How to get IBM Redbooks” on page 132. Note that some of the documents referenced here may be available in softcopy only.

- ▶ *DB2 UDB for z/OS Version 8: Everything You Ever Wanted to Know, ..and More*, SG24-6079
- ▶ *Ready for e-Business: OS/390 Security Server Enhancements*, SG24-5158
- ▶ *Multilevel Security and DB2 Row Level Security Revealed*, SG24-6480
- ▶ *DB2 UDB for z/OS Version 8 Technical Preview*, SG24-6871

## Other publications

These publications are also relevant as further information sources:

- ▶ *DB2 UDB for z/OS V8 RACF Access Control Module Guide*, SC18-7433
- ▶ *z/OS Security Server RACF Systems Programmer's Guide*, SA22-7681
- ▶ *z/OS Security Server RACF Macros and Interfaces*, SA22-7682
- ▶ *z/OS Security Server RACF Systems Administrator's Guide*, SA22-7683
- ▶ *z/OS Security Server RACF Command Language Reference*, SA22-7687
- ▶ *z/OS Security Server RACF Callable Services*, SA22-7691
- ▶ *z/OS Security Server RACF RACROUTE Macro Reference*, SA22-7692
- ▶ *z/OS Cryptographic Services PKI Services Guide and Reference*, SA22-7693
- ▶ *z/OS Security Server RACF Diagnosis Guide*, GA22-7689
- ▶ *z/OS LDAP Server Administration and Use*, SC24-5923
- ▶ *z/OS UNIX System Services Planning*, SA22-7801
- ▶ *z/OS Planning for Multilevel Security and Common Criteria*, GA22-7509
- ▶ *IBM Health Checker for z/OS User's Guide*, SA22-7994

## Online resources

These Web sites and URLs are also relevant as further information sources:

- ▶ RACF/DB2 conversion utility  
<http://www.ibm.com/servers/eserver/zseries/zos/racf/racfdb2.html>
- ▶ LDAP browser editor is freely downloadable from:  
<http://www-unix.mcs.anl.gov/~gawor/ldap/>

## How to get IBM Redbooks

You can search for, view, or download Redbooks, Redpapers, Hints and Tips, draft publications and Additional materials, as well as order hardcopy Redbooks or CD-ROMs, at this Web site:

[ibm.com/redbooks](http://ibm.com/redbooks)

## Help from IBM

IBM Support and downloads

[ibm.com/support](http://ibm.com/support)

IBM Global Services

[ibm.com/services](http://ibm.com/services)



# Index

## A

ACEE  
    XAPLACEE 49  
ACTIVATE parameter  
    synchronized copy 23  
ADDUSER command  
    pass phrase 11  
administrative authorities 44  
advanced encryption standard 2  
AES 2  
ALTUSER command  
    pass phrase 12  
APAR OA12443  
    sharing RACF database 122  
APAR OA1491  
    SPE for V1R7 28  
APD 15  
ARRANGE command 84  
AUTODIRECT.target-node.USER.PHRSSYNC resource  
    profile 14  
automatic password direction 15

## B

BPXPRMxx MAXFILEPROC command 130

## C

CA domain 67  
CDTINFO segment 123  
CERTAUTH virtual key ring 59–60  
certificate authority 65  
change log 112  
CK command 80, 82  
CK panel in SDSF 80

## D

DB2 authority checking 33  
DB2 authorization checks 33  
DB2 catalog 41  
DB2 database authorities 45  
DB2 internal security 40  
DB2 object privileges 42  
DB2 objects 38  
    RACF groups 44  
DB2 security 4  
digital certificate support 4  
DIGTNMAP class 58  
DIGTRING class 58  
DSMON utility 19  
DSNX@XAC exit 34  
DSNX@XAC exit point 48  
DSNXRXAC module 35  
DSNXSXAC module 35

## E

event code qualifier 35 18  
event code qualifier 7 18

## F

FASTAUTH check  
    IRRSDA00 ACEE create 5  
FTP.DATA file 60

## G

GDBM  
    change log backend 110  
GDBM backend directory 112  
GDSNTB  
    RACF grouping class 42  
GRANT statement 41  
group change logging 3, 112

## H

HTTP protocol  
    SCEP certificates 4  
HZSPRMxx parmlib member 78, 80, 85–86, 89, 91–92

## I

IBM Tivoli Directory Integrator 116  
ICHERCDE macro 123  
ICHPWX11 exit 9–11, 19  
identity mapping  
    DB2 users 4  
IDS 2  
IDS policy solution 2  
IKYSETUP REXX EXEC 68  
Intrusion Detection Services 2  
inventory control block 121  
IRR.PASSWORD.RESET resource profile 17  
IRR912I message 41  
IRRADU00 utility 18  
IRRDBU00 utility 4, 19  
IRRDPI00 LIST command 127  
IRRMING00 utility 120  
IRRRID00 utility 48  
IRRSDA00 utility 5  
IRRSIDL00 59  
IRRUT200 utility 3, 22  
    copy database 22  
    PARM=ACTIVATE 24  
    synchronized copy 22  
IRRUT400 utility 3

## K

Kerberos principle names 128

KERBLINK class 128–129  
KERBLINK resource class 128  
key ring 58  
    password enveloping 114  
KEYRING directive 60

## L

LDAP 110  
LDAP change log 3  
LDAP server 2  
LISTUSER command 4, 113  
    pass phrase 12  
LOCKINPUT option  
    IRRUT400 3

## M

MDSNTB  
    RACF member class 42  
MDSNTB resource class 39  
multilevel security 52, 56

## P

pass phrase 8  
    rules 9  
pass phrase exit  
    ICHPWX11 9  
pass phrase history 10  
password 8  
PASSWORD command 13  
password envelopes 113–114  
password phrase 2, 7–8, 10–13, 17  
password phrase auditing 17  
password synchronization 15  
PASSWORD.ENVELOPE profile 115  
PHRASE keyword 13  
PHRASE operand 9  
PHRASESYNC resource profile 14  
PHRDATE 19  
PHRGEN 19  
PKCS #7 format 114  
PKI 64  
PKI Services 4  
pkiprereg utility 69, 73  
pkiserv.conf file 73  
PKIX standard 64  
Policy filters 90  
primary authorization ID 41  
protecting DB2 object 40  
public key infrastructure 64  
PWSYNC resource profile 14

## R

R\_datalib callable service  
    IRRSDL00 58–59  
RACDCERT command 58–60  
RACF  
    DSNXRXAC exit 31  
    GENERIC OWNER facility 32

IRR@XACS exit 31–32  
RACF access control module 34, 43  
    customizing and activating 36  
RACF access control module source  
    DSNXSXAC and DSNXRXAC 35  
RACF report writer 52  
RACF/DB2 resource classes 37  
RACFEVNT class 113, 115–116  
RACFEVNT resource class profiles 113  
RALTER command  
    GENERIC keyword 124  
RDEFINE CDTINFO command 123  
Redbooks Web site 132  
    Contact us x  
RLIST command  
    CDTINFO segment 124  
root certificat 58  
RRSFDATA class  
    user ID associations 14  
RRSFDATA profiles 14  
RVARY ACTIVE command 22  
    synchronized copy 23  
RVARY INACT command 23

## S

SAF identity token 4  
SCEP 4, 68  
SCEP messages 4  
SCEP support 4  
SDBM backend 112–113  
SECDATA resource class 55  
Secure Sockets Layer 58  
security label  
    defining 55  
SET PWSYNC command  
    pass phrase synchronization 14  
SETROPTS PASSWORD INTERVAL suboperand 10  
SETROPTS PASSWORD(HISTORY) command 10  
SETROPTS PASSWORD(INTERVAL) value 10  
shared posit numbers 125  
Simple Certificate Enrollment Protocol 68  
simple certificate enrollment protocol 4  
SMF data unload utility 52  
SSL 58  
synchronized copy  
    RACF database 22  
SYS1.SAMPLIB  
    IRR@XACS 32–33

## T

TAPEVOL class 3  
    TAPEDSN option 2  
TDBM backend 110  
TDBM entry 112

## U

USER class 58

## **V**

virtual key ring 4, 59

## **X**

XFACILIT class 79–80

## **Z**

z/OS LDAP 112

Archived

Archived









# z/OS Version 1 Release 8 RACF Implementation



**Password phrase,  
RACF DB2 security,  
RACF health checks**

**RACF virtual key ring,  
PKI Services, LDAP  
change logging**

**Template extensions,  
IRRUT200, IRRUT400  
utilities**

This IBM Redbook describes the implementation of RACF in z/OS Version 1 Release 8. This release continues to deliver industry leadership for security. Improvements have been introduced to further enhance the security-rich environment z/OS users rely on. These enhancements include:

- ▶ RACF support for virtual key rings to treat the collection of all the certificates owned by one user ID, including the SITE and CERTAUTH reserved user IDs, as an independent key ring. This will eliminate the need to manually create multiple real key rings for SSL-enabled z/OS client applications such as FTP.
- ▶ RACF template extensions allow templates to expand beyond their current 4K size.
- ▶ RACF supports the use of passwords longer than eight characters, now called password phrases.
- ▶ The RACF access control module exit, DSNXRAC, has changed substantially with DB2 version 9. A RACF administrator can now define a security rule before an object is created and preserve the rule for a dropped object. In addition, RACF general resources for member and group profiles can be used by an installation to protect multiple DB2 resources with a single RACF profile.
- ▶ A new parameter on the IRRUT200 utility tells the utility to activate the backup data set printed to as output. This is accomplished by the utility internally issuing an RVARV ACTIVE for the backup data set after the copy is complete. IRRUT200 and IRRUT400 utilities now check whether their output data sets are active primary or backup RACF data sets on this system.
- ▶ New RACF health checks are introduced.
- ▶ RACF in z/OS V1R8 provides a solution to some functional gaps in the way that change logging of RACF profile updates were reflected in z/OS LDAP, and an enhancement is made to LISTUSER to demonstrate whether password enveloping is enabled for a user.

In addition to describing the new features, this book includes detailed steps for implementing these enhancements. It explains how to configure them for your installation and how to use them to increase the security of your environment.

**INTERNATIONAL  
TECHNICAL  
SUPPORT  
ORGANIZATION**

**BUILDING TECHNICAL  
INFORMATION BASED ON  
PRACTICAL EXPERIENCE**

IBM Redbooks are developed by the IBM International Technical Support Organization. Experts from IBM, Customers and Partners from around the world create timely technical information based on realistic scenarios. Specific recommendations are provided to help you implement IT solutions more effectively in your environment.

**For more information:  
[ibm.com/redbooks](http://ibm.com/redbooks)**

SG24-7248-00

ISBN 0738489859