

z/OS



Security Server RACF Auditor's Guide

Version 2 Release 1

Note

Before using this information and the product it supports, read the information in "Notices" on page 195.

This edition applies to version 2, release 1, modification 0 of IBM z/OS (product number 5650-ZOS) and subsequent releases and modifications until otherwise indicated in new editions.

© **Copyright IBM Corporation 1994, 2013.**

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Figures	v
--------------------------	----------

Tables	vii
-------------------------	------------

About this document ix

Intended audience	ix
How to use this document	ix
Where to find more information	ix
RACF courses	x
Other sources of information	x
Internet sources	x

How to send your comments to IBM xiii

If you have a technical problem	xiii
---	------

z/OS Version 2 Release 1 summary of changes xv

Chapter 1. The RACF auditor. 1

AUDITOR and group-AUDITOR attribute	1
Access control and accountability	1
Logging	3
Owner-controlled logging	4
Auditor-controlled logging.	5
Using the RACF cross-reference utility program (IRRUT100).	7
Using the RACF database unload utility program (IRRDBU00)	7
Using the RACF SMF data unload utility program (IRRADU00)	8
Using the DFSORT ICETOOL.	8
Using the RACF report writer	9
Conducting the audit	9
Preliminary information	9
System information.	10
RACF implementation.	12

Chapter 2. Setting audit controls 17

Auditing for z/OS UNIX System Services	32
Classes that control auditing for z/OS UNIX System Services	33
Auditable events	36
Commands	36
Audit options for file and directory levels	36
Auditing for superuser authority in the UNIXPRIV class	37
Auditing for the RACF remote sharing facility (RRSF)	38

Chapter 3. The RACF SMF data unload utility 49

Operational considerations	49
Using IRRADU00	49

Writing your own application	51
IRRADU00 example	51
Return codes for IRRADU00.	52
IRRADU00 output	52
Using output from the RACF SMF data unload utility	53
Sort/Merge programs	53
Relational databases	53
XML.	53
Using the DFSORT ICETOOL to create reports.	54
The report format	54
The record selection criteria	55
Using the RACFICE PROC to generate reports	55
Reports based on the SMF data unload utility (IRRADU00)	56
Creating customized reports.	57
Using the RACF SMF data unload utility output with DB2	58
Steps for using IRRADU00 output with DB2	58
Creating a DB2 database for unloaded RACF SMF data	59
Creating a DB2 table space	59
Creating the DB2 tables	60
Loading the DB2 tables	61
Reorganizing the unloaded RACF SMF data in the DB2 database	63
Creating optimization statistics for the DB2 database	63
Deleting data from the DB2 database	63
DB2 table names.	63
Using the RACF SMF data unload utility to generate XML documents.	65
XML overview	66
Producing XML output	67
How the XML tag names are derived.	68
Viewing and working with XML audit reports.	69
Event code qualifiers	69
Event 1(1): JOB INITIATION/TSO LOGON/TSO LOGOFF	70
Event 2(2): RESOURCE ACCESS	74
Event 3(3): ADDVOL/CHGVOL	76
Event 4(4): RENAME RESOURCE	77
Event 5(5): DELETE RESOURCE	78
Event 6(6): DELETE ONE VOLUME OF A MULTIVOLUME RESOURCE	78
Event 7(7): DEFINE RESOURCE	79
Event 8(8)–25(19): COMMANDS	80
Event 26(1A): APPCLU	81
Event 27(1B): GENERAL AUDITING	82
Event 28(1C)–58(3A): z/OS UNIX EVENT TYPES	82
Event 59(3B): RACLINK EVENT TYPES	85
Event 60(3C)–62(3E): z/OS UNIX XPG4 EVENT TYPES	85
Event 63(3F): z/OS UNIX SETGROUPS EVENT TYPE	86

Event 64(40): X/OPEN SINGLE UNIX SPECIFICATION EVENT TYPES	86
Event 65(41): z/OS UNIX PASSING OF ACCESS RIGHTS EVENT TYPES	86
Event 66(42)–67(43): CERTIFICATE EVENT TYPES	86
Event 68(44): GRANT OF INITIAL KERBEROS TICKET	87
Event 69(45): R_PKIServ GENCERT	87
Event 70(46): R_PKIServ EXPORT	87
Event 71(47): POLICY DIRECTOR ACCESS CONTROL DECISION.	87
Event 72(48): R_PKIServ QUERY	88
Event 73(49): R_PKIServ UPDATEREQ	88
Event 74(4A): R_PKIServ UPDATECERT.	88
Event 75(4B): CHANGE FILE ACL	88
Event 76(4C): REMOVE FILE ACL.	88
Event 77(4D): SET FILE SECURITY LABEL	88
Event 78(4E): SET WRITE-DOWN PRIVILEGE.	89
Event 79(4F): CRL PUBLICATION.	89
Event 80(50): R_PKIServ RESPOND	89
Event 81(51): PassTicket Evaluation	89
Event 82(52): PassTicket Generation	89
Event 83(53): R_PKIServ SCEPREQ	89
Event 84(54): R_Datalib RDATAUPD	89
Event 85(55): PKIAURNW	90
Event 86(56): R_PgmSignVer.	90
Event 87(57): RACMAP	90
Event 88(58): AUTOPROF	90
Event 89(59): RPKIQREC	90

Audit function codes for z/OS UNIX System Services	90
--	----

Chapter 4. The data security monitor (DSMON) 95

Appendix A. The RACF report writer 119

Sample reports	152
Merging SMF records produced by RACF for z/VM with SMF records produced by RACF for MVS	172

Appendix B. XML Schema 173

Appendix C. Accessibility 191

Accessibility features	191
Using assistive technologies	191
Keyboard navigation of the user interface	191
Dotted decimal syntax diagrams	191

Notices 195

Policy for unsupported hardware.	196
Minimum supported hardware	197
RSA Secure code	197
Trademarks	197

Index 199

Figures

1. Sample ISPF panel for RACF	7	25. Sample RACF Started Procedures Table Report (STARTED Class Active)	112
2. GLOBALAUDIT Operand on the ALTDSD Command	30	26. Selected User Attribute Report	114
3. JCL to Invoke the RACF SMF data unload utility	52	27. Selected User Attribute Summary Report	114
4. DFSORT's ICETOOL Utility	54	28. Sample Selected Data Sets Report	118
5. Member SELU: Selected User Report report format statements	55	29. RACF Report Writer Overview.	120
6. Member SELUCNTL: Selected User Report record selection statements	55	30. JCL for Dumping SMF Records and Running the Report Writer as a Batch Job	125
7. Report for all IRRADU00 Records Associated with a Specific User ID.	56	31. Key to Symbols in Command Definitions	128
8. Customized Record Selection Criteria.. . . .	57	32. Summary Activity Report from SMF	153
9. Customized Report Format	58	33. Standard Header Page (1/3)	154
10. Customized Report JCL	58	34. Standard Header Page (2/3)	155
11. Sample SQL Utility Statements Defining a Table Space	59	35. Standard Header Page (3/3)	156
12. Sample SQL Utility Statements Creating a Table.	61	36. General Summary Report	157
13. DB2 Utility Statements Required to Load the Tables	62	37. Listing of Status Records.	158
14. DB2 Utility Statements Required to Delete the Group Records	63	38. Listing of Process Records	159
15. Specifying DSMON JCL	96	39. Short User Summary Report	160
16. Reports produced by DSMON	100	40. Short Group Summary Report	160
17. Sample System Report	102	41. Short Resource Summary Report	161
18. Sample Group Tree Report	102	42. Short Command Summary Report	162
19. Sample Program Properties Table Report	104	43. Short Event Summary Report	163
20. Sample RACF Authorized Caller Table Report	105	44. Short Owner Summary Report.	164
21. Class Descriptor Table Report	107	45. User by Resource Summary Report	164
22. Sample RACF Exits Report	108	46. Group by Resource Summary Report	165
23. Sample RACF Global Access Checking Table Report	109	47. Resource by User Summary Report	165
24. Sample RACF Started Procedures Table Report (ICHRIN03).	111	48. Resource by Group Summary Report	166
		49. Resource by Event Summary Report	167
		50. Event by Resource Summary Report	168
		51. Command by User Summary Report	169
		52. Command by Group Summary Report	170
		53. Command by Resource Summary Report	171
		54. Owner by Resource Summary Report	172

Tables

1. How RACF Simulates DB2 Authorization Checking	45	8. XML special characters substitutions	69
2. LOGSTR Data.	46	9. Audit Function Codes for z/OS UNIX System Services.	90
3. Return codes for the SMF data unload utility (IRRADU00)	52	10. Reports Specified by the FUNCTION Control Statement	99
4. ICETOOL Reports from IRRADU00 Output	56	11. Reports Specified by the USEROPT Control Statement	99
5. Correlation of DB2 Table Names and Record Types	63	12. Summary of RACFRW Command and Its Operands	123
6. XML naming exceptions	68	13. Summary of RACFRW Subcommands	123
7. XML interpretation of special characters example.	68		

About this document

This document supports z/OS® (5650-ZOS) and contains information about the Resource Access Control Facility (RACF®), which is part of the Security Server. The Security Server works in conjunction with these components:

- Integrated Security Services components
- Open Cryptographic Enhanced Plug-Ins
- PKI Services
- Resource Access Control Facility (RACF)
- z/OS Firewall Technologies
- z/OS LDAP Server
- z/OS Security Server Network Authentication Service

Note that some of the components referenced are not part of Security Server but are included in other z/OS packages.

This document describes the role of the RACF auditor and explains the auditing tools that RACF provides. Reports on system and resource use can provide the auditor with information about the basic system-security environment of an installation.

If you need specific information about using RACF on z/VM® systems, refer to the RACF Version 1 documentation. Information describing how to use RACF in a shared database environment with z/OS and z/VM systems (for example, shared database function and templates in support of database unload) remains in RACF documentation.

Intended audience

This document is intended for those individuals defined as RACF auditors (persons who have the AUDITOR or group-AUDITOR user attribute).

You should be familiar with both RACF and z/OS, or z/OS and z/VM if you are running RACF on two or more systems that share the same RACF database.

How to use this document

This document provides detailed information about the RACF SMF data unload utility, which allows you to create a sequential file from the security relevant audit data. Also, chapters cover the RACF report writer, the data security monitor (DSMON), and optional audit controls for tracking RACF events.

Where to find more information

When possible, this information uses cross-document links that go directly to the topic in reference using shortened versions of the document title. For complete titles and order numbers of the documents for all products that are part of z/OS, see *z/OS Information Roadmap*.

To find the complete z/OS library, including the z/OS Information Center, see z/OS Internet Library (<http://www.ibm.com/systems/z/os/zos/bkserv/>).

RACF courses

The following RACF classroom courses are available in the United States:

ES191 *Basics of z/OS RACF Administration*

BE870 *Effective RACF Administration*

ES885 *Exploiting the Advanced Features of RACF*

IBM® provides various educational offerings for RACF. For more information about classroom courses and other offerings, do any of the following:

- See your IBM representative
- Call 1-800-IBM-TEACH (1-800-426-8322)

Other sources of information

IBM provides customer-accessible discussion areas where RACF may be discussed by customer and IBM participants. Other information is also available through the Internet.

Internet sources

The following resources are available through the Internet to provide additional information about the RACF library and other security-related topics:

- **Online library**

To view and print online versions of the z/OS publications, use this address:

<http://www.ibm.com/systems/z/os/zos/bkserv/>

- **Redbooks®**

The documents known as IBM Redbooks that are produced by the International Technical Support Organization (ITSO) are available at the following address:

<http://www.redbooks.ibm.com>

- **Enterprise systems security**

For more information about security on the S/390® platform, and z/OS, including the elements that comprise the Security Server, use this address:

<http://www.ibm.com/systems/z/advantages/security/>

- **RACF home page**

You can visit the RACF home page on the World Wide Web using this address:

<http://www.ibm.com/systems/z/os/zos/features/racf/>

- **RACF-L discussion list**

Customers and IBM participants may also discuss RACF on the RACF-L discussion list. RACF-L is not operated or sponsored by IBM; it is run by the University of Georgia.

To subscribe to the RACF-L discussion and receive postings, send a note to:

listserv@listserv.uga.edu

Include the following line in the body of the note, substituting your first name and last name as indicated:

```
subscribe racf-l first_name last_name
```

To post a question or response to RACF-L, send a note, including an appropriate

Subject: line, to:

racf-l@listserv.uga.edu

- **Sample code**

You can get sample code, internally-developed tools, and exits to help you use RACF. This code works in our environment, at the time we make it available, but is not officially supported. Each tool or sample has a README file that describes the tool or sample and any restrictions on its use.

To access this code from a Web browser, go to the RACF home page and select the “Resources” file tab, then select “Downloads” from the list, or go to <http://www-03.ibm.com/systems/z/os/zos/features/racf/goodies.html>.

The code is also available from [ftp.software.ibm.com](ftp://software.ibm.com) through anonymous FTP. To get access:

1. Log in as user **anonymous**.
2. Change the directory, as follows, to find the subdirectories that contain the sample code or tool you want to download:

```
cd eserver/zseries/zos/racf/
```

An announcement will be posted on the RACF-L discussion list whenever something is added.

Note: Some Web browsers and some FTP clients (especially those using a graphical interface) might have problems using [ftp.software.ibm.com](ftp://software.ibm.com) because of inconsistencies in the way they implement the FTP protocols. If you have problems, you can try the following:

- Try to get access by using a Web browser and the links from the RACF home page.
- Use a different FTP client. If necessary, use a client that is based on command line interfaces instead of graphical interfaces.
- If your FTP client has configuration parameters for the type of remote system, configure it as UNIX instead of MVS™.

Restrictions

Because the sample code and tools are not officially supported,

- There are no guaranteed enhancements.
- No APARs can be accepted.

How to send your comments to IBM

We appreciate your input on this publication. Feel free to comment on the clarity, accuracy, and completeness of the information or provide any other feedback that you have.

Use one of the following methods to send your comments:

1. Send an email to mhvrcfs@us.ibm.com.
2. Send an email from the "Contact us" web page for z/OS (<http://www.ibm.com/systems/z/os/zos/webqs.html>).
3. Mail the comments to the following address:
IBM Corporation
Attention: MHVRCFS Reader Comments
Department H6MA, Building 707
2455 South Road
Poughkeepsie, NY 12601-5400
US
4. Fax the comments to us, as follows:
From the United States and Canada: 1+845+432-9405
From all other countries: Your international access code +1+845+432-9405

Include the following information:

- Your name and address.
- Your email address.
- Your telephone or fax number.
- The publication title and order number:
z/OS V2R1.0 Security Server RACF Auditor's Guide
SA23-2290
- The topic and page number that is related to your comment.
- The text of your comment.

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute the comments in any way appropriate without incurring any obligation to you.

IBM or any other organizations use the personal information that you supply to contact you only about the issues that you submit.

If you have a technical problem

Do not use the feedback methods that are listed for sending comments. Instead, take one of the following actions:

- Contact your IBM service representative.
- Call IBM technical support.
- Visit the IBM Support Portal at z/OS support page (<http://www.ibm.com/systems/z/support/>).

z/OS Version 2 Release 1 summary of changes

See the following publications for all enhancements to z/OS Version 2 Release 1 (V2R1):

- *z/OS Migration*
- *z/OS Planning for Installation*
- *z/OS Summary of Message and Interface Changes*
- *z/OS Introduction and Release Guide*

Chapter 1. The RACF auditor

RACF is a flexible security tool. It allows you to set your own security objectives and use RACF to help achieve those objectives in a way that best meets your installation's needs.

Although installations might have slightly different security needs, certain RACF user roles or tasks are common to all users. At any installation, different users have different levels of responsibility for security or different needs to access resources. Some people might have extensive responsibility for security, whereas others might have little or none; some users might require almost unlimited access to resources, whereas others might need only limited access, and some might be barred from entering the system at all.

The primary means of defining a user's responsibility for security is the RACF *user attribute*. A user attribute is a part of the RACF definition of what an installation allows a particular user to do. The SPECIAL attribute, for example, is normally assigned to the RACF security administrator; a SPECIAL user can execute any RACF command except those reserved for a user with the AUDITOR attribute.

This separation of powers is necessary because it is the security administrator's job to establish RACF controls; it is the auditor's job to test the adequacy and effectiveness of these controls. In this sense, your job as the auditor is very similar to the job of a financial auditor in a bank.

AUDITOR and group-AUDITOR attribute

Once a SPECIAL user assigns the AUDITOR user attribute to you, your responsibility is to verify that RACF is meeting your installation's security goals. As a RACF auditor, your job is essentially the same, regardless of whether you have the AUDITOR attribute (with responsibility for checking RACF controls on a user, or system-wide, level) or the group-AUDITOR attribute (with responsibility for checking RACF controls for a group and its subgroups). Whereas a user with the group-AUDITOR attribute can only monitor the users and resources owned by a specific group and its subgroups, the responsibility is so much like that of a user with the AUDITOR attribute that this document applies to both and notes any specific differences.

Access control and accountability

As the auditor, you are responsible for checking that RACF is meeting the installation's needs for access control and accountability. Access control means that you can control user accesses to resources and verify that the accesses allowed are appropriate to the particular resource. For example, you might question why a tape librarian had access to a payroll data set. The auditor must verify that an installation has a way to maintain accountability. Accountability means that you can trace activities on the protected system to a particular person. Normally, several people should not share a user ID. A user ID can be shared by people who use a digital certificate to identify and authenticate themselves. In this case, accountability is maintained because each person's unique X500 name is audited in addition to the shared user ID.

Attention:

If the person responsible for setting or resetting passwords uses the NOEXPIRED keyword of the ALTUSER command to set a new, unexpired password for a user, you might experience problems maintaining accountability. For example, an administrator might reset an expired password with the following command:

```
ALTUSER VIOLA PASSWORD(ME51NOW)
```

When VIOLA uses the password for the first time, RACF forces her to change it. After she changes the password, only VIOLA knows the new password, which provides reasonable assurance that the audit record indicates that VIOLA performed some action.

However, the administrator might issue:

```
ALTUSER VIOLA PASSWORD(ME51NOW) NOEXPIRED
```

VIOLA gets an unexpired password and does not need to change it, which means that the administrator and VIOLA both know the password. Although an audit record written by the ALTUSER command indicates whether NOEXPIRED was specified, there is no assurance that an audit event with VIOLA in the record occurred because of something VIOLA did. You need to consider the possibility that the administrator performed the action that caused RACF to write the audit record.

To help you to audit access control and accountability, RACF provides:

- Logging routines that record the information you require
- Audit control functions that enable you to specify the information RACF is to record (or log)
- The RACF SMF data unload utility, which converts SMF records into formats which can be used by a relational database manager, such as an XML version which can be easily viewed by a web browser
- The DFSORT ICETOOL, which generates reports from RACF SMF data unload utility information and RACF database unload utility information
- The data security monitor (DSMON), which generates reports containing information about the security environment for MVS
- The RACF report writer, which generates tailored reports based on the information you have directed RACF to log

To specify the audit control functions, use either the RACF ISPF panels or the RACF commands to direct RACF to log any events relevant to your installation's data security program.

After RACF has logged security events, you can analyze this log by:

- Loading the records produced by the RACF SMF data unload utility into a relational database manager for analysis.
- Creating XML output of the report and viewing the results in a web browser. This report can also be customized by the use of an XSLT style sheet file.
- Invoking the RACF report writer to print the data RACF has logged and use the reports to identify possible security violations or weaknesses in the security mechanism.

The data security monitor (DSMON) generates a set of reports that lets you audit the current status of the data security environment. You can use the information in the reports to compare the actual system characteristics and resource protection

levels with the installation's requirements. If the installation has not defined ICHDSM00 (DSMON) as a controlled program, you must have the AUDITOR attribute to run DSMON.

If DSMON has been defined as a controlled program, you must have at least EXECUTE or READ access to the ICHDSM00 resource in the PROGRAM class. Additionally, with READ access from the PROGRAM class if DSMON is not running in a clean program-controlled environment you also need the AUDITOR attribute.

Logging

Logging, the recording of data about specific events, is the key to auditing the use of RACF at your installation. You must ensure that RACF logs the information you need. RACF uses the system management facilities (SMF) to log data about various RACF events. RACF writes SMF records to an SMF data set or log stream.



Things to Consider

- Each additional logging activity that you specify increases RACF and SMF processing and, as a result, might affect RACF performance.
- When RACF is enabled for sysplex communication, RACF logs the use of commands only for the system from which the command originated (if auditing has been turned on), even though some commands are propagated to the other members in the RACF sysplex data sharing group.
- When you are sharing a RACF database among two or more systems, you must run the logging and reporting utilities from the highest level system.

RACF always logs information about certain events because knowing about these events is essential to an effective data-security mechanism. The events that RACF always logs are:

- Every use of the RVARY or SETROPTS command.
If you are using the RACF subsystem on MVS and issue RVARY as an MVS operator command, the job name information is propagated in the SMF record. This distinguishes it from an RVARY command issued from a TSO session.
- Every time a RACROUTE REQUEST=VERIFY request fails or an initACEE fails because a certificate is unknown or not trusted.
- Every time a distributed identity is unknown.
- Every time the console operator grants access to a resource as part of the failsoft processing performed when RACF is inactive
- When a user not defined as a z/OS UNIX System Services user tries to dub a process

- When an unauthorized user tries to mount or unmount the file system
- When a user successfully sets or resets his write-down mode, or fails attempting to do so because the user does not have the write-down privilege
- Other components may also cause security events to be logged

For more details about z/OS UNIX System Services events for which audit records are always written, see *z/OS UNIX System Services Planning*.

RACF never logs some events, because knowing about these events is not essential to effective data security. RACF never logs any use of the following RACF commands: LISTDSD, LISTGRP, LISTUSER, RLIST, and SEARCH.

In addition, RACF can *optionally* log other events. Optional logging is under the control of either a resource-profile owner or the auditor.

Owner-controlled logging

Owners of resources can specify, in the resource profile, what types of accesses to log (successes, failures, or both) and what level of access to log (READ, UPDATE, CONTROL, or ALTER). Owners can also specify that no logging is to occur for an access that is a success or failure. Owner-controlled logging is not directly under your control, but you should verify that resource owners request a level of logging that is consistent with the sensitivity of the resource. Furthermore, your installation can use three methods to *override* the logging that an owner specifies in the resource profile.

1. First, you can suppress auditing for all resources in a specific class by specifying LOGOPTIONS(NEVER(*class-name*)) on the SETROPTS command. Likewise, you can activate auditing for all access attempts for all resources in a specific class by specifying LOGOPTIONS(ALWAYS(*class-name*)). See “Activating auditing for access attempts by class” on page 21.
2. Second, if you have the AUDITOR attribute, you can specify additional logging that supersedes the owner's logging specification for a specific resource by *adding* audit controls to the resource profile. Note that you cannot *change* the owner's logging specifications for a specific resource profile, only add to them. You can do this for specific resource profiles by specifying the GLOBALAUDIT operand on the ALTDSD or RALTER command. The use of these controls is described in “Data set controls” on page 29 and “General resource controls” on page 30.
3. Third, for resources that have their authority checked by RACROUTE REQUEST=AUTH, your installation can bypass a profile owner's logging specification by using the RACROUTE REQUEST=AUTH postprocessing exit routine. This exit routine can, for certain accesses, specify unconditional logging or unconditionally suppress logging. For example,
 - An installation might use the exit routine to specify unconditional logging for accesses to a highly classified resource.
 - An installation might suppress logging when the exit routine recognizes READ access to common system resources, such as SYS1.MACLIB.

You should be aware of any such exit-routine specifications. For more information about using exit routines, see *z/OS Security Server RACF System Programmer's Guide*.

Note to z/OS UNIX System Services Users

Owner-controlled logging for z/OS UNIX files is specified in the file security packet (FSP) instead of a profile. The access levels are different and logging is set with the **chaudit** command. For more information about this command, see *z/OS UNIX System Services User's Guide*.

Auditor-controlled logging

You can direct RACF to log additional events. These events are:

- Changes to any RACF profiles
- All RACF commands issued by users who either had the SPECIAL attribute, or gained authority to issue the command because they had the group-SPECIAL attribute
- All unauthorized attempts to use RACF commands
- All RACF-related activities of specific users
- All accesses to resources (data sets and general resources) that RACF allows because the user has the OPERATIONS or group-OPERATIONS attribute
- All accesses to specific data sets
- All accesses to specific general resources
- All accesses to resources protected by specific profiles in the SECLABEL class
- All accesses to a specified class of resources at an access level indicated on the LOGOPTIONS keyword of the SETROPTS command
- Selected events in related APPC/MVS transactions
- z/OS UNIX System Services events. See Chapter 3, "The RACF SMF data unload utility," on page 49 for event codes and a table of event code qualifiers.

You can identify which of these events apply to your installation's security goals and use audit controls to direct RACF to log the events you require.

Choosing between using RACF TSO commands and ISPF panels

In general, you can perform the same RACF functions using RACF TSO commands and ISPF panels.

The **RACF TSO commands** provide the following advantages:

- Entering commands can be faster than displaying many panels in sequence.
- Using commands from the documented examples is more straightforward. (The examples in the RACF documents are generally command examples.)
- Getting online help for RACF TSO commands

You can get online help for the RACF TSO commands documented in *z/OS Security Server RACF Command Language Reference*.

- To see online help for the PERMIT command, for example, enter:

```
HELP PERMIT
```

- To limit the information displayed, specify operands on the HELP command. For example, to see only the syntax of the PERMIT command, enter:

```
HELP PERMIT SYNTAX
```

Restriction: TSO online help is not available when RACF commands are entered as RACF operator commands.

- Getting message ID information

If a RACF TSO command fails, you receive a message. If you do not get a message ID, enter:

```
PROFILE MSGID
```

Reenter the RACF TSO command that failed. The message appears with the message ID. See *z/OS Security Server RACF Messages and Codes* for help if the message ID starts with ICH or IRR.

Restriction: PROFILE MSGID cannot be entered as a RACF operator command.

The **ISPF panels** provide the following advantages:

- When you use the panels, you avoid having to memorize a command and type it correctly. Panels can be especially useful if the command is complex or you perform a task infrequently.
- ISPF creates in the ISPF log a summary record of the work that you do. Unless you use the TSO session manager, the RACF commands do not create such a record.
- From the panels, you can press the HELP key to display brief descriptions of the fields on the panels.
- The options chosen when installing the RACF panels determine whether output (for example, profile listings, search results, and RACF options) is displayed in a scrollable form.
- The ISPF panels for working with password rules allow you to enter all of the password rules on one panel. Figure 1 on page 7 shows one of these panels.
- When you use the ISPF panels to update a custom field definition in the CFDEF segment, the current values are displayed. You can then overwrite the values to make changes.
- When you use the ISPF panels to add, update, or delete custom field information (CSDATA segment fields) in a user or group profile, the panels are primed with the custom field names and values. You can then make additions, changes, and deletions.

Limitations: The following limitations apply to the use of the ISPF panels:

- The ISPF panels do not support all options of all commands. For example, the SETROPTS PASSWORD option to activate and deactivate mixed-case password support is not available through the RACF panels.
- The ISPF RACF panels are limited to 32000 lines of command output. If the output listing for a command (most commonly, the RLIST command) exceeds 32000 lines, the output is truncated at the 32000 line limit and an error is likely to occur. To avoid this limitation, use one of the following alternate methods:
 - Issue the command using a batch execution of the terminal monitor program (TMP) and use the SDSF XD command to store the output in a data set.
 - Create a report using output from the RACF database unload (IRRDBU00) utility.

```

RACF - SET PASSWORD FORMAT RULES
COMMAND ==>

Enter PASSWORD FORMAT RULES:
          MINIMUM  MAXIMUM
          LENGTH   LENGTH   FORMAT
RULE 1:  ---      ---      -----
RULE 2:  ---      ---      -----
RULE 3:  ---      ---      -----
RULE 4:  ---      ---      -----
RULE 5:  ---      ---      -----
RULE 6:  ---      ---      -----
RULE 7:  ---      ---      -----
RULE 8:  ---      ---      -----
To cancel an existing rule, enter NO for MINIMUM LENGTH.
To specify FORMAT, use the following codes for each character position:
* = Any Character      $ = National      V = Vowel      N = Numeric
C = Consonant         A = Alphabetic   v = Mixed Vowel  m = Mixed Numeric
c = Mixed Consonant   L = Alphanumeric W = No Vowel

```

Figure 1. Sample ISPF panel for RACF

Using the RACF cross-reference utility program (IRRUT100)

If you have the AUDITOR or SPECIAL attribute, you can use the RACF cross-reference utility to find and list occurrences of a user ID or group name in the RACF database.

If you have the group-AUDITOR or group-SPECIAL attribute, you can use these utilities only for a user ID or group that is within your scope of authority.

You can also process your profile or profiles that you own.

Remember

Before using the RACF cross-reference utility, you should consult with your RACF system programmer. You may need to find out:

- *How* to run the utility
- *When* to run the utility to reduce its impact on system operations

For more information about using this utility, see *z/OS Security Server RACF System Programmer's Guide*.

Using the RACF database unload utility program (IRRDBU00)

You can also use the RACF database unload utility to provide flexibility in analyzing RACF profile information. The output from this utility is a sequential file that is a relational representation of a RACF database.

If the output is loaded into a database management system (such as DB2*), you can issue your own queries. For example:

- You can find and list occurrences of a user ID or group name in the RACF database
- You can list members of a group by name rather than user ID
- You can list the last recorded date and time that a RACROUTE REQUEST=VERIFY request was issued for a user.

A user with the SPECIAL attribute can request that RACF record statistics during RACROUTE REQUEST=VERIFY processing. REQUEST=VERIFY is issued when a user is logging on to a system or a batch job is entering a system including when RACF does such work as a directed command, application update, or password change on behalf of the user. See *z/OS Security Server RACF Security Administrator's Guide* for more information about recording RACROUTE REQUEST=VERIFY statistics.

Before using the RACF database unload utility, you should consult with your RACF system programmer. You may need to find out *how* to run the utility. Your input database must be in the correct format and you must have UPDATE authority to it.

For more information about running this utility, see *z/OS Security Server RACF Macros and Interfaces* and *z/OS Security Server RACF Security Administrator's Guide*. For information about using this utility with DB2®, see *z/OS Security Server RACF Security Administrator's Guide*.

Using the RACF SMF data unload utility program (IRRADU00)

The RACF SMF data unload utility program is the recommended utility for processing RACF audit records. With it, you can create a sequential file from the security relevant audit data. You can use the sequential file in several ways. You can:

- View the file directly
- Use the file as input for installation-written programs
- Manipulate the file with sort/merge utilities
- Browse an XML-formatted output

You can also upload the file to a database manager (for example, DB2) to process complex inquiries and create installation-tailored reports.

For details about the RACF SMF data unload utility program, see Chapter 3, “The RACF SMF data unload utility,” on page 49.

Using the DFSORT ICETOOL

IBM's DFSORT product provides a reporting facility called ICETOOL. RACF provides a collection of reports in IRRICE, a member in SYS1.SAMPLIB, which you can use to create your own reports. IRRICE uses DFSORT statements for the selection criteria and ICETOOL statements for the report format for all the reports. The IEBUPDTE utility processes the IRRICE member and creates a partitioned data set that contains the report formats and record selection criteria.

If you want to use the ICETOOL to create RACF reports, you must:

- Be sure you have the IBM's DFSORT product or its equivalent installed on your system
- Unpack the DFSORT ICETOOL control statements that are supplied by RACF
- Customize the DFSORT ICETOOL control statements supplied by RACF to produce the reports you need

See “Using the DFSORT ICETOOL to create reports” on page 54 for a detailed description of the DFSORT ICETOOL and the IRRICE member.

Using the RACF report writer

The profile listings the RACF commands provide can help you to verify the audit controls that exist at any particular time. The RACF report writer helps you to monitor RACF-related activity during system operation and to verify that these activities are consistent with your installation's security goals. It provides printed reports based on the data your audit controls directed RACF to log.

The report writer makes use of certain system management facility (SMF) records to obtain information. You can control the selection of these records and the format and type of report that the report writer produces through the use of the RACFRW command and its subcommands.

However, the report writer supports audit records for RACF release 1.9.2 and earlier. It does not support most of the audit records introduced in the RACF Version 2 releases or as part of the z/OS releases.

See Appendix A, “The RACF report writer,” on page 119 for a detailed description of the report writer, the RACFRW command, and samples of the available reports.

Conducting the audit

Asking the right questions is an essential part of *any* audit, including an audit of your own RACF-protected installation or a review of another installation. In such a review or audit, your principal review objectives are:

1. Judge how effectively RACF has been implemented to handle security at the installation.
2. Identify any security exposures.
3. Recommend ways to improve the system.

To accomplish these objectives, you need to understand your installation and its security requirements. To obtain the information, you can interview a few key people such as the security administrator, the system programmer responsible for installing and implementing RACF, and a senior member of the system support group. Asking the right questions of the right people can help you in your audit.

One way to deal with the mass of information used for an audit is to divide it into categories: preliminary information, system information, and RACF information. The rest of this chapter uses these categories to identify blocks of information you need or questions you might ask. Not all of the suggestions apply at any one installation; any particular installation may require additional investigation. Treat these suggestions as a starting point, then tailor and expand your audit to fit the conditions that exist.

When you are conducting an audit, you should obtain current installation reports from the data security monitor (DSMON). These reports are helpful in answering a number of your questions. You can also use the DSMON reports to verify that the *actual* status of various security mechanisms is what you and the installation expect. DSMON is described in Chapter 4, “The data security monitor (DSMON),” on page 95.

Preliminary information

Before conducting an audit, you should establish preliminary information concerning the type, size, and complexity of your installation. The following questions should help you get started.

1. What are the processor complexes and their associated system control programs (SCPs), including the release and level of RACF for each? You can use the DSMON reports to answer this particular question.
2. For each processor complex, what are the subsystems—such as TSO/E, IMS/ESA[®], CICS/ESA—protected by RACF (including the release and level of each)? List them.
3. Are processor complexes linked (for example, by NJE, RSCS, JES2, or JES3)?
4. Is DASD shared between systems? What type of data is shared?
5. Do you have dial-up lines?
6. Explain briefly the classification system.
7. What is the highest classification of data processed and/or transmitted?
8. Will you be using z/OS UNIX System Services?

System information

An operating system should have integrity; that is, it should prevent one program from interfering with or modifying the execution of another system or user program unless the interference is authorized. To increase your awareness of potential security problems, read related MVS documentation that provide overview information and describe system features that promote security. A list of the related documentation is provided in the preface of this document.

Basic system

Use the following questions to help establish foundation information concerning your system.

1. What is the operating system version and release level and PTF level (PUT tape)? You can use the DSMON reports to answer this particular question.
2. How many local modifications have been applied (excluding exit routines)?
3. What are the main areas and/or functions modified?
4. Are the systems the same on all processor complexes?
5. What user-written SVC routines does the system include and what is their purpose?
6. What exit routines are in the system and what is their purpose? Could these exit routines affect RACF protection? Some examples of subsystems or components that can have exit routines are:

SMF

TSO/E

JES

Job management

Authorization

Use the following questions to determine current system authorization.

1. What are the entries in the program properties table (PPT) that automatically bypass password protection? You can use the DSMON reports to answer this particular question.
2. Which started procedures have the trusted or privileged attribute? You can use the DSMON reports to answer this particular question.
3. What are the authorized libraries?
 - In your PARMLIB concatenation (IEAAPFxx)? You can use the DSMON reports to answer this particular question.

- In your PARMLIB concatenation (LNKLSTxx)? You can use the DSMON reports to answer this particular question.
- In your PARMLIB concatenation (IEALPAxx)?
- In your PARMLIB concatenation (LPALSTxx)?

Note: You can find your PARMLIB concatenation with an MVS operator command or you can use the RACF_SENSITIVE_RESOURCES health check which reports on the concatenated PARMLIB data sets.

4. Other than standard IBM programs, what programs require authorization in these libraries?
5. What are the commands and programs that can be executed in the foreground as Authorized Program Facility (APF)-authorized (CSECTs IKJEFTE2 and IKJEFTE8 in module IKJEFT01 or IKJTABLS, or SYS1.PARMLIB member IKJT000, depending on your release of TSO)?
6. Is the list of authorized programs and commands reasonable and consistent with the installation's security goals? You can use the DSMON reports to answer this particular question.
7. How are changes and additions to the authorized libraries controlled? Who authorizes changes?

System protection

Use the following questions to determine current system protection.

1. How are changes to the system controlled and documented?
2. How are the system libraries (including page data sets, dump data sets, JES spool and checkpoint data sets, and SMP data sets) protected? Who can access these libraries?
3. What libraries have a universal access of READ? You can use the DSMON reports to answer this particular question.
4. What libraries have a universal access of UPDATE or higher? You can use the DSMON reports to answer this particular question.
5. What libraries have a universal access of EXECUTE? You can use the DSMON reports to answer this particular question.
6. Are the DLIB data sets also protected? You can use the DSMON reports to answer this particular question.
7. Are all the catalogs protected? You can use the DSMON reports to answer this particular question.
8. Are key security items, (such as RACF databases, SYS1.UADS, password data, cipher key file, SMF data, source and load modules for RACF exit routines, and SMF routines) all identified and protected? You can use the DSMON reports to answer this particular question.
9. If JES3 is installed, is the use of dynamic support program (DSP) controlled (including utilities such as tape-to-tape and tape-to-print)?

Miscellaneous

The following questions do not fall into any of the preceding categories; however, the information gained from the answers could be useful when conducting an audit.

1. Can bypass label processing (BLP) be used? If yes, how is it controlled?
2. Is OS password protection used? If yes, why?
3. If dial-up terminals are used, how is unauthorized use prevented?

4. Is full SMF recording in use? If not, what is excluded either by options or exit routine code?
5. What is the wait limit that causes a terminal to be logged off?
6. How far back do system backup dumps go?
7. Are all IPLs logged and the reasons reported?
8. Is all time on the system accounted for?
9. Is it possible to detect if the system has been loaded without RACF? You can use the DSMON reports to answer this particular question.
10. How is the use of RACF commands controlled?

RACF implementation

Installing RACF does not necessarily mean that the RACF security facilities were correctly implemented and are being correctly maintained. (For more information about implementing RACF, see *z/OS Security Server RACF Security Administrator's Guide*.)

Protection plan

Use the following questions to determine what resources your installation is currently protecting.

1. How many RACF users and groups do you have? All or part of this question can be answered by using the DSMON reports.
2. Do you have any non-RACF users? If so, why?
3. Which of the following resources are RACF-protected, what proportion of each is protected, and how is it decided which to protect? All or part of this question can be answered by manipulating the output of the RACF database unload utility.

DASD data sets

Tape data sets

Nodes

Terminals

IMS/ESA

CICS/ESA

DB2 resources

Programs

Surrogate user IDs

TSO procedures

TSO account numbers

Unit record devices

Graphics devices

TP devices

Operator commands

VTAM[®] applications

JES writers

JES SYSIN and SYSOUT data sets

Job names

JES input devices

MCS consoles

Temporary data sets

Hiperbatch

User IDs that cannot be propagated

TSO message transmission

Key resources unique to the installation

z/OS UNIX System Services resources

4. How does the installation ensure that appropriate protection is maintained?
5. What protection is available for resources *not* protected by RACF?
6. Is the protection policy reasonable?

Usage

Use the following questions to determine how RACF is currently being implemented.

1. Which user IDs (including started tasks) have any of the following privileged attributes or authorities? Why? You can use the IRRICE reports or DSMON reports to answer this particular question.
 - SPECIAL and group-SPECIAL
 - OPERATIONS and group-OPERATIONS
 - AUDITOR and group-AUDITOR
 - CLAUTH
 - JOIN
 - CONNECT
 - GRPACC
2. How is the granting of these privileges controlled?
3. Are user IDs shared? If so, why, and how is accountability maintained? Is the RESTRICTED attribute used to limit the resource access of the shared user IDs?
4. Is the default for UACC always NONE? If not, why?

All or part of this question can be answered by manipulating the output of the RACF database unload utility or by using the sample reports contained in the IRRICE member of SYS1.SAMPLIB.
5. How are password qualities complied with? Do you use, for example, password length, nature (alphabetic, alphanumeric, no vowels), repetition, or change frequency?
6. What RACF information, such as the following, is logged to SMF?
 - Command violations
 - Changes to profiles
 - Accesses to specific resources
 - Actions of SPECIAL and group-SPECIAL users
 - Actions of OPERATIONS and group-OPERATIONS users
7. Who decides what resource-access information is to be collected? On what criteria?
8. What RACF statistics are collected?
9. What are the access rules when RACF is inactive or unavailable, such as stopping production, performing repair work only, or allowing selected jobs and applications to run?
10. Is WARNING mode active, entirely or partially? Are there non-WARNING mode resources?

All or part of this question can be answered by manipulating the output of the RACF database unload utility.

11. Do access lists contain groups rather than individuals?
12. How is the authority to run production work handled? Does the job submitter have access to production data? If so, how are the profiles deleted?
13. How is RACF protection handled in disaster-recovery plans?
14. Describe any operational or usage problems for which the installation cannot currently determine a solution.
15. Do you need to delete tape profiles before using tape volumes again?
16. Is DASDVOL authorization used instead of the OPERATIONS user attribute?

Technical

The following questions provide technical orientation.

1. What RACF exit routines are used, and what functions do they perform? The following list identifies the exits. You can use the DSMON reports to answer this particular question.

-

Exit Routine

Function

ICHDEX01

password authentication

ICHDEX11

password authentication

ICHRIX01

RACROUTE REQUEST=VERIFY preprocessing

ICHRIX02

RACROUTE REQUEST=VERIFY postprocessing

ICHRCX01

RACROUTE REQUEST=AUTH preprocessing

ICHRCX02

RACROUTE REQUEST=AUTH postprocessing

ICHRDX01

RACROUTE REQUEST=DEFINE preprocessing

ICHRDX02

RACROUTE REQUEST=DEFINE postprocessing

ICHCCX00

command preprocessing

ICHCNX00

command preprocessing

ICHRFX01

RACROUTE REQUEST=FASTAUTH preprocessing

ICHRFX02

RACROUTE REQUEST=FASTAUTH postprocessing

ICHRFX03

RACROUTE REQUEST=FASTAUTH preprocessing

ICHRFX04

RACROUTE REQUEST=FASTAUTH postprocessing

ICHPWX01

new password

ICHPWX11

new password phrase

ICHRLX01

RACROUTE REQUEST=LIST pre/postprocessing

ICHRLX02

RACROUTE REQUEST=LIST selection

ICHRSMFE

report writer

IRRACX01

ACEE compression and expansion

IRRACX02

ACEE compression and expansion

IRRE VX01

command pre/postprocessing

IRR VAF01

custom field validation exit

2. How are the exit routine functions and changes authorized and controlled?
3. Who is allowed to update exit routine code (both source and load form)?
4. What SETROPTS options are used? Are any important protection or monitoring functions set off?
5. Have basic RACF facilities been enhanced, excluding exit routine code?
6. How many primary RACF databases are there? You can use the DSMON reports to answer this particular question.
7. Does each primary RACF database have a backup on a different volume? You can use the DSMON reports to answer this particular question.
8. What other backup facilities exist for RACF databases?
9. How is the RACF database synchronized after a restore?
10. Are all RACF databases adequately protected, and who has access to them? You can use the DSMON reports to answer this particular question.
11. How does the installation control the switching and deactivating of the RACF databases (RVARY command, IPL/database name table)?
12. Are any special checks required on the use of PERMIT?
13. How are passwords and password phrases protected against disclosure when batch jobs are submitted through internal readers?
14. How are restores of entire volumes handled? How are synchronization problems between volumes and the RACF databases resolved?
15. What are the RACF class names as defined in the class descriptor table? What are the UACCs associated with these names? Can OPERATIONS users access the resources by default? You can use the DSMON reports to answer this particular question.
16. Is there a global access table, and what resources are specified in the table? You can use the DSMON reports to answer this particular question.
17. What is in the started procedures table (ICHRIN03), and is the authority of the associated user IDs appropriate? You can use the DSMON reports to answer this particular question.

Administration control

The following questions provide information concerning how RACF is administered at your installation.

1. Who is responsible for the administration of RACF? You can use the DSMON reports to answer this particular question.
2. Who is responsible for the technical aspects of RACF?
3. Are data owners identified?
4. Do data owners classify their data?
5. Is the degree of protection provided by the installation based on the owner classification?
6. Are there written and approved procedures for RACF administration?
7. Does the installation maintain written records of requests for changes to RACF protection and the resulting actions taken?
8. How are users and groups administered? How are additions, deletions, changes, connections, and authorities handled?
9. How is the authority to protect resources and grant access checked and handled?
10. How is the granting of temporary authorities handled? Can users issue PERMIT/CONNECT for temporary access, or are there privileged attributes available for emergency use?
11. How is password distribution handled?
12. How are lost passwords handled?
13. Is additional verification required for users with privileged attributes? Are these users restricted to particular terminals?
14. Is there an emergency user ID with the SPECIAL attribute available for use when no other SPECIAL user ID can be used? If so, how does the installation protect the user ID and its password? You can use the DSMON reports to answer this particular question.
15. Is the auditor a different person from the RACF security administrator? What are the responsibilities of the auditor? You can use the DSMON reports to answer this particular question.
16. Is there any user education available?
17. Are there any entries in the authorized caller table? If so, why are they there and are they adequately protected?

Management control

The following questions address management control.

1. What reports are available to users, owners, and installation management to ensure that the system is not being misused? Examples are reports that identify violation attempts, unauthorized access attempts, and unauthorized use of commands and privileges.
2. How frequently are reports produced, and who sees them?
3. If a security violation occurs, what follow-up action does the installation take?
4. Is the installation using DSMON reports to monitor the basic system security environment? If not, why isn't it?

Chapter 2. Setting audit controls

Audit controls are special RACF functions that RACF allows only the auditor to perform. To preserve the checks and balances necessary to an effective security mechanism, not even the security administrator with the SPECIAL attribute can execute auditor functions. Therefore, you should ensure that SPECIAL users do not also have the AUDITOR attribute.

Audit control	Action
General	You can use: <ul style="list-style-type: none">• Auditing options specified on the SETROPTS (set RACF options) command
Specific	You can specify: <ul style="list-style-type: none">• All RACF related activities of specific users• Attempts to access data sets protected by specific profiles• Attempts to access general resources (such as terminals) that are protected by specific profiles

Some audit controls are product-specific. Refer to the appropriate product documentation for setting these audit controls.

General audit controls

You specify general (system-wide) audit controls on either the SETROPTS command or the SET AUDIT OPTIONS ISPF panel. General audit controls direct RACF to log (or not to log) certain security-relevant events, such as the activities of OPERATIONS or group-OPERATIONS users, RACF command violations, and attempts to access RACF-protected resources.

To specify the general audit controls, you must have the AUDITOR attribute. After you have initially established your controls or modified existing controls, it is a good practice to list the current options to verify that the controls are correct.

If you have the AUDITOR attribute, you can specify these SETROPTS operands or request the function on the corresponding panel:

APPLAUDIT and NOAPPLAUDIT
AUDIT and NOAUDIT
CMDVIOL and NOCMDVIOL
LIST
LOGOPTIONS
OPERAUDIT and NOOPERAUDIT
REFRESH GENERIC
REFRESH RACLIST
SAUDIT and NOSAUDIT
SECLABELAUDIT and NOSECLABELAUDIT
SECLEVELAUDIT and NOSECLEVELAUDIT

If you have the group-AUDITOR attribute, you can use only the LIST and REFRESH GENERIC operands.

Logging RACF commands and DEFINE requests

If you have the AUDITOR attribute, you can specify the classes for which RACF logs all detected accesses to the RACF database through RACF commands and DEFINE requests. You can specify this option with the AUDIT operand on the SETROPTS command; it becomes effective immediately. The following example specifies that you want RACF to log RACF commands and DEFINE requests for users, groups, data sets, and the TERMINAL general-resource classes.

```
SETROPTS AUDIT(USER GROUP DATASET TERMINAL)
```

If you specify AUDIT(*), RACF logs RACF command and DEFINE request activity for all classes.

If you want to log any change in RACF protection for IMS™, enter:

```
SETROPTS AUDIT(IMS)
```

The following table shows the events that SETROPTS AUDIT(*class*) affects:

User	Group	Data set	Classes in the CDT
ADDUSER	ADDGROUP	ADDSD	PERMIT
ALTUSER	ALTGROUP	ALTDSD	DEFINE Request
CONNECT	CONNECT	DELDSD	RALTER
DELUSER	DELGROUP	PERMIT	RDEFINE
getUMAP	getGMAP	DEFINE Request	RDELETE
initACEE registration / deregistration	REMOVE		
PASSWORD			
RACDCERT			
RACLINK			
RACMAP			
REMOVE			
R_pkiserv			
VERIFY			

If you have the AUDITOR attribute, you can also specify the NOAUDIT operand on the SETROPTS command and identify the class or classes for which you do not want RACF to log RACF command and DEFINE requests. If you specify NOAUDIT(*), RACF does not log RACF commands and DEFINE requests for any class.

NOAUDIT(*) is in effect at RACF initialization.

Note: If you have the AUDITOR attribute, you can specify with the UAUDIT operand on the ALTUSER command that you want RACF to log the following:

- All RACF commands (except LISTDSD, LISTGRP, LISTUSER, RLIST, and SEARCH) issued by this user
- All additions, changes, or deletions that the user makes to RACF profiles using RACROUTE REQUEST=DEFINE requests
- All attempts that the user makes to access RACF-protected resources, except those authorized by global access checking and those not logged because the

resource manager (issuer of the RACROUTE REQUEST=AUTH or RACROUTE REQUEST=FASTAUTH request) specified no logging

- All security decisions that are made during RACF callable services involving this user and any resource in certain z/OS UNIX classes. For a list of these classes, see “Auditing for z/OS UNIX System Services” on page 32.

Bypassing logging of activity of users with the SPECIAL attribute

If you have the AUDITOR attribute, you can request that RACF bypass logging of all RACF commands and the AUTH and DEFINE requests issued by users with the SPECIAL or group-SPECIAL attribute. You can specify this option with the NOSAUDIT operand on the SETROPTS command as shown in the following example:

```
SETROPTS NOSAUDIT
```

If you have the AUDITOR attribute, you can also specify the SAUDIT operand on the SETROPTS command, to indicate that you want RACF to log the command and request activity (except LISTDSD, LISTGRP, LISTUSER, RLIST, and SEARCH, which are never logged) of users with the SPECIAL or group-SPECIAL attribute.

Note: If you are concerned only with how SPECIAL users change profiles, you do not need to specify SAUDIT if AUDIT(*) is in effect.

SAUDIT is in effect at RACF initialization.

Logging the activities of users with the OPERATIONS attribute

If you have the AUDITOR attribute, you can audit all accesses to resources granted because the user has the OPERATIONS or group-OPERATIONS attribute, by using the OPERAUDIT operand on the SETROPTS command. The following example shows how to specify this option.

```
SETROPTS OPERAUDIT
```

If you specify OPERAUDIT, RACF logs all accesses to RACF-protected resources granted because the user has the OPERATIONS or group-OPERATIONS attribute, and all uses of the ADDSD, and RDEFINE commands allowed because a user has the OPERATIONS or group-OPERATIONS attribute.

Note: Some programs that call RACF functions such as RACROUTE REQUEST=AUTH and RACROUTE REQUEST=DEFINE can request that RACF perform no logging. Thus, if an OPERATIONS or group-operations user accesses a protected resource through such a program, RACF does not log the access even if you request OPERAUDIT.

OPERAUDIT overrides the audit field of data set, file, directory and general resource profiles. OPERAUDIT does not affect any auditing requested by the GLOBALAUDIT operand on the RACF commands.

If you have the AUDITOR attribute, you can also specify NOOPERAUDIT. NOOPERAUDIT does no special auditing of users with the OPERATIONS or group-OPERATIONS attribute.

NOOPERAUDIT is in effect at RACF initialization.

Logging and bypassing RACF command violations

A violation can occur because RACF does not authorize a user to modify a particular profile or to enter a particular operand on a command.

If you have the AUDITOR attribute, you can specify the CMDVIOL operand on the SETROPTS command. This operand tells RACF to log all command violations (except for LISTDSD, LISTGRP, LISTUSER, RLIST, and SEARCH, which are never logged).

Note: Specifying CMDVIOL causes RACF to log all the command violations that it detects. You can then use the RACF report writer to produce a printed audit trail of command violations. You can determine how many command violations are occurring and which users are causing the violations. A significant number of command violations, especially when RACF is first installed, may indicate the need for more user education. The report can also help you to identify any specific users who are persistently trying to alter profiles without the proper authority.

CMDVIOL is in effect at RACF initialization.

If you have the AUDITOR attribute, you can request that RACF bypass logging of all violations detected by RACF commands (except RVARY and SETROPTS, which are always logged) during RACF command processing. You can specify this option with the NOCMDVIOL operand on the SETROPTS command as shown in the following example:

```
SETROPTS NOCMDVIOL
```

Activating auditing for security levels

If you have the AUDITOR attribute, you can activate auditing of access attempts to all RACF-protected resources. To activate this option, specify the SECLEVELAUDIT operand with an installation-defined security level name on the SETROPTS command. Auditing is done if the profile protecting a resource is equal to or greater than the security level you specify on the SECLEVELAUDIT operand.

Note:

1. You can only specify a security level name defined by your installation in the SECLEVEL profile in the SECDATA class. If you specify a security level that is not in the SECLEVEL profile for the SECDATA class, RACF ignores the operand and does no logging.
2. The SECDATA class must be active if you want RACF to perform security level control.

The following example shows how to activate auditing based on the security level CONFIDENTIAL. (This example assumes that the installation has defined the level CONFIDENTIAL in the SECLEVEL profile.)

```
SETROPTS SECLEVELAUDIT(CONFIDENTIAL)
```

When you specify a security level, RACF audits all attempts to access resources with the specified security level and higher. This option allows your installation to audit access attempts to a RACF-protected resource, based on the sensitivity of the resource, as determined by the installation. If you do not specify a security level, RACF audits all access attempts to all resources for which your installation has defined a security level (SECLEVEL).

Note:

1. If a program issues an AUTH or DEFINE request and specifies that RACF should not perform any logging, RACF does not log the event even if you request logging.
2. When RACF grants access to a resource because of an entry in the global access checking table, RACF does not log the event even if you request logging.

If you have the AUDITOR attribute, you can also deactivate auditing of access attempts to RACF-protected resources based on installation-defined security levels. To deactivate this option, specify the NOSECLEVELAUDIT operand on the SETROPTS command.

NOSECLEVELAUDIT is in effect at RACF initialization.

Activating auditing for access attempts by class

If you have the AUDITOR attribute, you can audit attempts to access resources in specified classes according to the option selected. You can specify the DATASET class and any active classes in the class descriptor table. The resources need not have profiles created in order for the auditing to occur.

The following command specifies that auditing be done for all attempts to access the TERMINAL class.

```
SETROPTS LOGOPTIONS(ALWAYS(TERMINAL))
```

In this case, auditing is done every time a user logs on at any terminal on the system, whether that terminal is protected by a profile or not, and whether that profile specifies auditing or not.

You can specify that auditing be done for the following conditions:

ALWAYS

All attempts to access resources protected by the class are audited.

NEVER

No attempts to access resources protected by the class are audited. (All auditing is suppressed.)

SUCSESSES

All successful attempts to access resources protected by the class are audited.

FAILURES

All failed attempts to access resources protected by the class are audited.

DEFAULT

Auditing is controlled by the profile protecting the resource, if a profile exists. You can specify DEFAULT for all classes by specifying an asterisk (*) with DEFAULT.

Note:

1. The SUCSESSES and FAILURES operands result in auditing in addition to any auditing specified in profiles in the class. In contrast, the ALWAYS and NEVER operands override any auditing specified in profiles in the class.
2. If LOG=NONE is specified on a RACROUTE REQUEST=AUTH, it takes precedence and auditing is not performed.

3. When RACF grants access to a resource because of an entry in the global access checking table, RACF does not log the event even if you request logging.
4. If authority checking is performed with a RACROUTE REQUEST=FASTAUTH request, auditing is not affected by a SETROPTS LOGOPTIONS command.

LOGOPTIONS(DEFAULT(*)) is in effect at RACF initialization.

If your installation has specified SETROPTS LOGOPTIONS for any number of classes and you want this reset, specify LOGOPTIONS(DEFAULT(*)) on the SETROPTS command.

Activating auditing for security labels

If you have the AUDITOR attribute, you can audit all attempts to access resources whose profiles have a security label specified. The auditing that is done is specified in the SECLABEL profile that defines the security label. To do this, specify the SETROPTS command as follows:

```
SETROPTS SECLABELAUDIT
```

When SECLABELAUDIT is in effect, the SECLABEL profiles for which RACLIST processing has been done enhance the auditing specified in resource profiles. For example, if the security label EAGLE has been defined by the installation and a resource with security label EAGLE is accessed, when a user with security label EAGLE logs on, RACF records the event if either:

- The in-storage copy of the SECLABEL profile named EAGLE requires it, or
- The profile protecting the resource requires it.

For example, to audit all failed accesses to resources with a security label of EAGLE, the installation should issue the following command:

```
RALTER SECLABEL EAGLE AUDIT(FAILURES(READ))
```

After this command has been issued, a DATASET profile that has a security label of EAGLE, but no auditing specified, will have failed access attempts audited due to the security label auditing specified.

Note: A value of NONE in the SECLABEL profile does not suppress auditing; auditing is determined by other auditing specifications (such as the resource profile).

NOSECLABELAUDIT is in effect at RACF initialization.

If your installation has specified SETROPTS SECLABELAUDIT, additional auditing is done based on SECLABEL profiles. This option can be reset to the default by specifying NOSECLABELAUDIT on the SETROPTS command. The auditing options in the SECLABEL profiles do not have to be changed, however, because NOSECLABELAUDIT causes the audit options to be ignored.

The SECLABELAUDIT function applies whenever resources are accessed or defined, and includes accessing and defining z/OS UNIX files and directories. SECLABELAUDIT is checked during the following operations:

- RACROUTE REQUEST=AUTH
- RACROUTE REQUEST=FASTAUTH
- RACROUTE REQUEST=DEFINE .

Additionally, SECLABELAUDIT is checked during the following file and directory operations:

- ck_access
- ck_IPC_access
- ck_owner_two_files
- make_FSP
- make_ISP
- ck_process_owner
- R_ptrace.

Note that auditing is determined not only by the security label of a resource, but also of the user. Therefore, if a resource's security label does not request auditing, but a user has a security label which does request auditing, auditing will be performed.

When auditing security labels with the SECLABELAUDIT function, SMF audit records are written, thus requiring a high amount of system overhead. It is advised that auditing **not** be turned on for every security label in the system. Only those security labels with specific auditing requirements, as defined by the installation, should be audited.

Auditing for APPC/MVS

There are several considerations associated with APPC/MVS auditing:

- Auditing user verification requests as transactions enter the system and complete
- Auditing the use of a particular APPC/MVS transaction program
- Determining the relationship between the audit records created during the execution of APPC/MVS transactions

User verification requests

There are two alternatives used in APPC/MVS that affect how auditing is performed. The alternative in effect is determined by the level of conversation security established between a pair of LUs. With either alternative, you can request a pair of audit records that mark the creation and deletion of a user's security environment.

1. One alternative uses a concept known as *persistent verification (PV)*. When PV is used, the security environment for a user is created when the user's *first* transaction request enters the system. The security environment *persists* over multiple transactions before being deleted.

In terms of audit records for user verification, a user is audited twice at the most:

- First, when the user begins work on the system
 - Next, when the user signs off, regardless of how many transactions are submitted.
2. In the other alternative (non-PV), the user's security environment is created and deleted for *each* transaction the user requests.

In terms of audit records for user verification, every transaction a user submits may be audited. This can potentially produce a large volume of SMF records.

With either alternative, the audit records marking the creation and deletion of the security environment contain a common audit key that links the audit records together.

With either alternative, the auditing is controlled with the APPL profile and the APPLAUDIT operand of the SETROPTS command. See “Activating APPC/MVS auditing.”

Transaction program auditing

Auditing of resource access attempts is done as part of day-to-day operations set up by the auditor or profile-owner for your installation. This existing auditing also occurs for transaction programs, but with a slight difference in audit records.

The audit records created by a transaction program contain an audit key that can be used to link audit records together.

In the case of persistent verification (where user verification is audited only twice-at signon and signoff), the audit key links records to a particular user. In the case of non-PV, the audit key links records created for a single transaction request.

Relationship of APPC/MVS audit records

Audit records created for users and transaction programs may be linked by a common key. All APPC/MVS audit records contain an 8-byte key that may be used to link the beginning and ending records together.

Activating APPC/MVS auditing

APPLAUDIT is a RACF option that allows user verification auditing to occur at the beginning and ending of a user's transaction processing work. Activating this auditing requires two steps:

1. You must specify the APPLAUDIT operand on the SETROPTS command.
2. You must request auditing for the APPL profile associated with an APPC/MVS LU.

Issue the following command:

```
SETROPTS APPLAUDIT
```

In addition to setting APPLAUDIT on, you must also request auditing for the APPL profile.

For example, you could issue the following command:

```
RALTER APPL profile-name GLOBALAUDIT(ALL)
```

where *profile-name* is the name of the APPC/MVS LU.

Note: The security administrator must have previously activated the APPL class, defined the APPL profile, and issued a SETROPTS RACLIST for the class.

To turn on auditing for the profile in the APPL class, use any of the following operands on the RALTER command:

- AUDIT(ALL)
- AUDIT(SUCCESS)

- AUDIT(FAILURE)
- GLOBALAUDIT(ALL)
- GLOBALAUDIT(SUCCESS)
- GLOBALAUDIT(FAILURE)

Note: Remember to issue a SETROPTS RACLIST REFRESH for the APPL class.

Deactivating APPC/MVS auditing

To disable auditing of APPC transactions, users with the AUDITOR attribute should specify the SETROPTS command as follows:

```
SETROPTS NOAPPLAUDIT
```

NOAPPLAUDIT is in effect at RACF initialization.

Refreshing profiles

You can use the SETROPTS command to refresh profiles. This includes refreshing:

- In-storage generic profiles
- Profiles processed by SETROPTS RACLIST
- The global access table
- The program access table
- Shared systems

Refreshing in-storage generic profiles

You may want to use GENERIC REFRESH after changing the logging options in a generic profile that protects a specific data set, as described in “Specific audit controls” on page 28. However, extensive use of GENERIC REFRESH can adversely affect system performance.

You can refresh in-storage generic profiles by specifying both the GENERIC and REFRESH operands on the SETROPTS command. When you specify both GENERIC and REFRESH, you also specify one or more classes for which you want RACF to refresh in-storage generic profiles. This causes all the in-storage generic profiles within the specified general resource class (except those in the global access checking table) to be replaced with new copies from the RACF database. The following example shows how to refresh in-storage generic profiles for the DATASET and TERMINAL classes:

```
SETROPTS GENERIC(DATASET TERMINAL) REFRESH
```

Note that you must issue this command each time you want RACF to perform the refresh process.

If you specify GENERIC(*), RACF refreshes profile lists for the DATASET class and all active classes in the except group resource classes (such as GTERMINL and GDASDVOL). When you initiate the refresh procedure, RACF sets an indicator in the RACF communication vector table for the class(es) that you specified. After the indicator is set, RACF refreshes the profile lists the next time it invokes the generic-profile search routine.

If you specify NOGENERIC on the SETROPTS command, RACF stops using in-storage generic profile lists but does not immediately delete them. RACF deletes the profile lists at the end of the job or TSO session, or when you again specify

GENERIC. When you specify GENERIC, RACF rebuilds the profile lists. (If SETROPTS GENLIST has been used on your system, a copy of the generic profiles for the resource resides in common storage. You can also use REFRESH GENERIC to refresh these in-storage generic profiles.)

For classes RACLISTed by either the SETROPTS RACLIST command or RACROUTE REQUEST=LIST, generic including discrete profiles for the class must be refreshed. This process is described in the next section.

Refreshing RACLISTed profiles

If SETROPTS RACLIST has been used on your system, copies of the discrete and generic profiles for any resource within a general resource class reside in a data space and can be shared among users. SETR RACLIST(classname) REFRESH causes the data space to be replaced with another data space containing new copies of the discrete and generic profiles from the RACF database.

If SETROPTS RACLIST has been issued for a general resource class and you change the logging options for a general resource profile in the class, you may want to use the REFRESH option to refresh the profile.

The following example shows how to refresh SETROPTS RACLIST processing for the DASDVOL and TERMINAL classes.

```
SETROPTS RACLIST(DASDVOL TERMINAL) REFRESH
```

The RACROUTE REQUEST=FASTAUTH service routine works with in-storage profiles RACLISTed by the RACROUTE REQUEST=LIST macro with ENVIR=CREATE specified. To refresh those profiles, the application must delete them by using RACROUTE REQUEST=LIST,ENVIR=DELETE and then re-create them using RACROUTE REQUEST=LIST,ENVIR=CREATE again. However, if the GLOBAL=YES parameter is specified, a refresh is accomplished with SETR RACLIST(classname) REFRESH.

SETROPTS REFRESH processing on shared systems

If RACF is enabled for sysplex communication, the refresh operation for SETROPTS processing is propagated to all members of the RACF sysplex data sharing group.

Otherwise, the command applies only to the system (z/VM or MVS) on which you issue the SETROPTS command. If your installation has two or more systems sharing a RACF database, you must issue the SETROPTS command on all systems to have the refresh done on all systems.

However, if you do not perform a refresh (issue the SETROPTS command with the REFRESH option) on a system sharing a RACF database and that system needs to re-IPL, the refresh takes effect on that system when re-IPL is performed.

When you issue a SETROPTS REFRESH command, or one of the propagated RVARY commands (ACTIVE, INACTIVE, DATASHARE, NODATASHARE, SWITCH) from one member of a RACF sysplex data sharing group, the request is audited only on the system from which you issue the command, and only if auditing has been selected for that system. The request is not audited on the peer member systems (regardless of whether auditing has been selected).

For more details about SETROPTS commands that are propagated to all members of the RACF sysplex data sharing group, refer to *z/OS Security Server RACF Command Language Reference*.

Examples for setting audit controls using SETROPTS

The following examples show how to set system-wide audit controls by using the SETROPTS command.

Note: If you want to list the current system-wide audit controls set with the SETROPTS command, enter:

```
SETROPTS LIST
```

You can also use the LIST operand on the SETROPTS command; for example:

```
SETROPTS SAUDIT LIST
```

Example 1

To log any changes to the profiles in the USER, GROUP, DATASET, and DASDVOL classes, enter:

```
SETROPTS AUDIT(USER,GROUP,DATASET,DASDVOL)
```

Example 2

To log RACF commands issued by SPECIAL and group-SPECIAL users, enter:

```
SETROPTS SAUDIT
```

Example 3

To log all accesses to resources that users make as a result of the OPERATIONS attribute, enter:

```
SETROPTS OPERAUDIT
```

Example 4

To log all successful password changes (including password phrase changes), enter:

```
SETROPTS AUDIT(USER)
```

Example 5

To log all RACF command violations, enter:

```
SETROPTS CMDVIOL
```

Example 6

To log all attempts to access any resource with a security level of confidential or higher enter:

```
SETROPTS SECLEVELAUDIT(CONFIDENTIAL)
```

Example 7

To refresh the in-storage, generic data set profiles, enter:

```
SETROPTS REFRESH GENERIC(DATASET)
```

Note: You can combine these six examples into a single SETROPTS command by entering:

```
SETROPTS  AUDIT(USER,GROUP,DATASET,DASDVOL)
          SAUDIT OPERAUDIT CMDVIOL SECLEVELAUDIT(CONFIDENTIAL)
          REFRESH GENERIC(DATASET)
```

Example 8

To refresh the in-storage profiles for terminals when SETROPTS RACLIST has been used for the terminal class, enter:

```
SETROPTS  REFRESH RACLIST(TERMINAL)
```

Example 9

To log all device access checking for communication, unit record, and graphics devices, enter:

```
SETROPTS  LOGOPTIONS(ALWAYS(DEVICES))
```

Example 10

To log all operator commands that are protected by profiles in the OPERCMDS class, enter:

```
SETROPTS  LOGOPTIONS(ALWAYS(OPERCMS))
```

Example 11

To enable the use of SECLABEL profiles to determine the level of auditing you want, enter:

```
SETROPTS  SECLABELAUDIT
```

Example 12

To audit APPC transactions, enter:

```
SETROPTS  APPLAUDIT
RALTER  APPL profile-name AUDIT
```

where *profile-name* is the name of the APPC/MVS LU name.

Example 13 (z/OS UNIX System Services)

To log all failing directory searches and access checks for read/write access to directories, enter:

```
SETROPTS  LOGOPTIONS(FAILURES(DIRSRCH,DIRACC))
```

Example 14 (z/OS UNIX System Services)

To control auditing of the successful creation and deletion of file system objects and dubbing and undubbing of processes, enter:

```
SETROPTS  AUDIT(FSOBJ,PROCESS)
```

Specific audit controls

Specific audit controls enable you to log the following:

- All RACF-related activities for specific users
- Attempts to access specific data sets

- Attempts to access specific general resources
- Attempts to access resources protected by a security label

You can also list the complete contents of all profiles, including the owner-specified and auditor-specified logging options for resources.

If you have the AUDITOR attribute, you can set specific controls for any user, data set, or general resource, and list the contents of any profile. If you have the group-AUDITOR attribute, you can set controls and list profile contents only for those users, data sets, and general resources owned by the group in which you have the attribute, and any subgroup of that group.

User controls

You can use the UAUDIT or NOUAUDIT operand on the ALTUSER command, or request the corresponding functions on the AUDIT USER panel, to log all RACF-related activities for a specific user. When you set this control, RACF logs the following events:

- All RACF commands (except LISTDSD, LISTGRP, LISTUSER, RLIST and SEARCH) issued by this user
- All additions, changes, or deletions that the user makes to RACF profiles using RACROUTE REQUEST=DEFINE requests
- All attempts that the user makes to access RACF-protected resources, except those authorized by global access checking and those not logged because the resource manager (issuer of the RACROUTE REQUEST=AUTH or RACROUTE REQUEST=FASTAUTH request) specified no logging
- All security decisions made during RACF callable services involving this user and any resource in certain z/OS UNIX classes. For a list of these classes, see “Auditing for z/OS UNIX System Services” on page 32.

In general, you would probably not request user audit-logging as a matter of course, but it is useful in special situations. For example, you can specify user-audit logging if you suspect, based on other indicators such as command violations, that a particular user may be misusing the system or persistently trying to access or delete resources outside the user's control. Examples of the type of event that might indicate misuse of the system are either unauthorized attempts to modify a critical system resource (such as a PARMLIB data set) or a highly classified user resource (like payroll or business-planning data).

Example

To use the UAUDIT operand on the ALTUSER command to audit the person whose user ID is SMITH, enter:

```
ALTUSER SMITH UAUDIT
```

Data set controls

If owner controlled logging does not provide enough information for your audit, you can use the GLOBALAUDIT operand on the ALTDSD command or request the corresponding function on the AUDIT DATA SET ACCESS panel, in addition to the owner-specified logging values, to log user accesses to data sets.

GLOBALAUDIT allows you to specify logging for different kinds of attempts that users make to access resources at a given access level. With GLOBALAUDIT, you

can log successful accesses, failed accesses, or both to a given resource and specify READ, UPDATE, CONTROL, or ALTER for the access level to the resource.

Figure 2 summarizes the GLOBALAUDIT operand for ALTDSD and what you are able to specify for logging. (For a complete description of the ALTDSD command and its operands, see *z/OS Security Server RACF Command Language Reference*.)

```

      [           { ALL } ]
      [           {{{ FAILURES } } } ]
ALTDSD [ GLOBALAUDIT ( {{{ NONE } } (audit-access-level) } ... ) ]
      [           {{{ SUCCESS } } } ]

```

Figure 2. GLOBALAUDIT Operand on the ALTDSD Command

Note: Some authorized programs that call RACF to perform authority checking can request that RACF perform no logging. Therefore, if you request GLOBALAUDIT auditing for an access attempt made through such a program, RACF does not log the event.

As with the other specific controls, you do not audit accesses to most data sets, as a general rule. Therefore, GLOBALAUDIT(NONE) is the default for the operand. After you complete your audit of the data set, it is good practice to restore the default. When GLOBALAUDIT(NONE) is in effect, RACF logs accesses to the data set only as specified by the resource owner.

Example 1

To use the GLOBALAUDIT operand of the ALTDSD command to direct RACF to log all accesses to data set JIM.MEMO.TEXT, enter:

```
ALTDSD 'JIM.MEMO.TEXT' GLOBALAUDIT(ALL(READ))
```

Example 2

To use the GLOBALAUDIT operand of the ALTDSD command to direct RACF to log all failed accesses, all successful updates, and any scratch of data set A.B.C, enter:

```
ALTDSD 'A.B.C' GLOBALAUDIT(FAILURES(READ) SUCCESS(UPDATE))
```

General resource controls

You can use the GLOBALAUDIT operand on the RALTER command or request the corresponding function on the AUDIT GENERAL RESOURCES ACCESS panel to log user accesses to a specific general resource. Because the audit level that you specify on GLOBALAUDIT overrides the level the resource owner specified in the profile, you use it when the logging specified in the profile does not produce enough information for your needs.

When you set audit controls for a general resource, you specify what information RACF is to log—the result of the access attempt—and when RACF is to log the information—the level of access. Figure 2 shows the various valid combinations of what to log and when to log it.

As with the other specific controls, you would not audit accesses to most general resources usually. Therefore, GLOBALAUDIT(NONE) is the default for the operand. After you complete your audit of the general resource, it is good practice to restore the default. When GLOBALAUDIT(NONE) is in effect, RACF logs

accesses to the resource as specified in the profile.

Example

To use the RALTER command to specify auditing of all events for a tape volume NR1234, enter:

```
RALTER TAPEVOL NR1234 GLOBALAUDIT(ALL(READ))
```

Listing specific audit controls

RACF provides commands and corresponding ISPF panels that allow RACF users, depending on their authority or attributes, to examine the contents of RACF profiles. You, as auditor, can list the contents of all the RACF profiles (or all the profiles within the scope of your group if you are a group-AUDITOR). You can find a complete description of each of the commands, including sample output, in the *z/OS Security Server RACF Command Language Reference*.

The commands and the functions related to auditing are:

- **LISTDSD**

This lists the contents of data set profiles. If you have the AUDITOR attribute, you can list all profiles; if you have the group-AUDITOR attribute, you can list only those profiles within the scope of your group and its subgroups.

- **LISTGRP**

This lists the contents of group profiles. While the output does not contain any information directly related to specific audit controls, it does include information about the group structure and each user's authority within the group. This information may be useful to you. If you have the AUDITOR attribute, you can list all group profiles; if you have the group-AUDITOR attribute, you can list only the profiles within the scope of your group and its subgroups. This will not list all users in a universal group.

- **LISTUSER**

This lists the contents of user profiles. If you have the AUDITOR attribute, you can list all user profiles; if you have the group-AUDITOR attribute, you can list only those profiles within the scope of your group and its subgroups.

- **RLIST**

This lists the contents of general resource profiles. If you have the AUDITOR attribute, you can list all resource profiles; if you have the group-AUDITOR attribute, you can list only those profiles within the scope of your group and its subgroups.

Example

To list the complete profile for data set 'JIM.MEMO.TEXT', enter:

```
LISTDSD DA('JIM.MEMO.TEXT') ALL
```

Note: If no discrete profile exists for data set 'JIM.MEMO.TEXT', a generic profile may protect the data set. To list any such generic profile, enter:

```
LISTDSD DA('JIM.MEMO.TEXT') ALL GENERIC
```


Auditing for the RACF/DB2 external security module

The RACF/DB2 external security module allows you to use RACF resource profiles to check authorization for DB2 privileges and authorities. With these profiles, which represent the various DB2 privileges, you can use the RACF auditing tools to extract the information you need.

You can use the SMF data unload utility or the RACF report writer to extract and format the SMF records. When the RACF/DB2 external security module uses a RACROUTE REQUEST=FASTAUTH request to create an audit record, the record contains log string data that includes additional diagnosis information described in “Using the log string (LOGSTR) data” on page 46. You can use the log string information to link DB2 trace record IFCID 314 and a corresponding RACF SMF record.

Restriction

This topic contains information about using RACF with DB2 Version 7, and earlier DB2 versions. For information about using RACF with DB2 Version 8, and later DB2 versions, see <http://publib.boulder.ibm.com/infocenter/imzic>.

In addition, you can use the information found in messages IRR908I through IRR913I to help you understand how the RACF/DB2 external security module is set up for a particular subsystem. These messages identify the:

- Version and length of the RACF/DB2 external security module
- Name of the subsystem or group attach name
- FMID or APAR number associated with the module
- Customization options used for the module
- Classes that the module is trying to use
- Classes for which a RACROUTE request was successful
- &ERROROPT specifies the correct action to be taken for DB2 initialization and authorization errors.

Note: The system programmer sets these options. For detailed information, see *z/OS Security Server RACF System Programmer's Guide*.

Auditing for z/OS UNIX System Services

RACF writes audit records for the z/OS UNIX System Services auditable events in SMF type 80 records. The following classes are defined to control auditing:

- DIRSRCH
- DIRACC
- FSOBJ
- FSSEC
- IPCOBJ
- PROCESS
- PROCACT

The classes are in the class descriptor table (ICHRRCDX). No profiles can be defined in these classes. They are for audit purposes only. These classes do not need to be active to be used to control z/OS UNIX System Services auditing.

Activating the classes has no effect on auditing or authorization checking, except for the FSSEC class, which enables the use of ACLs in authorization checking.

Audit records are always written for security decisions made during RACF callable services involving resources in these z/OS UNIX classes when the user has the UAUDIT attribute, regardless of the LOGOPTIONS and AUDIT settings.

In addition, audit records are always written, and there is no option to turn them off, when one of the following conditions occurs:

- A user who is not defined as a z/OS UNIX System Services user tries to dub a process
- An unauthorized user tries to mount or unmount a file system

For more details about z/OS UNIX System Services events for which audit records are always written, see *z/OS UNIX System Services Planning*.

You can use profiles in the UNIXPRIV class to audit certain superuser functions. For more information about this z/OS UNIX System Services class, see “Auditing for superuser authority in the UNIXPRIV class” on page 37.

Classes that control auditing for z/OS UNIX System Services

Each of the classes controls auditing for z/OS UNIX System Services in a particular way. The descriptions that follow define the type of auditing each class controls and include:

- The audit event types that it controls
- The RACF callable services that write the audit record
- The z/OS UNIX services that can cause the event

The classes are:

DIRSRCH

Controls auditing of directory searches:

Audit event type:

28

RACF callable service:

ck_access

z/OS UNIX services:

chaudit, chdir, chmod, chmount, chmountsetuid, chown, getcwd, ioctl, lstat, link, mkdir, mknod, mount, mountsetuid, open, opendir, pathconf, readlink, rename, rmdir, stat, symlink, ttyname, unlink, unmount, unmountsetu, utime, chattr, vsetattr, vcreate, vmakedir, vlink, vremove, vrename, vsymlink, vresolvepn, vlookup, exec (indirectly using an open)

DIRACC

Controls auditing for access checks for read/write access to directories:

Audit event types:

29, 64

RACF callable service:

ck_access, ck_owner_two_files

z/OS UNIX services:

chmount, chmountsetuid, getcwd, ioctl, link, mkdir, mknod, mount, mountsetuid, open(new file), open(a directory), opendir,

remove, rename, rmdir, symlink, ttyname, unlink, unmount, unmountsetu, vlink, vmkdir, vcreate, vrename, vremovedir, vsymlink, vremove, vreaddir, utime (a directory)

FSOBJ

Controls auditing for all access checks for file system objects except directory searches using SETROPTS LOGOPTIONS and controls auditing of creation and deletion of file system objects using SETROPTS AUDIT (see note below).

For object access:

Audit event types:

30, 56

RACF callable service:

ck_access

z/OS UNIX services:

link, vlink, open, quiescesetu, unquiescesu, vreadwrite, utime, quiesce, unquiesce, exec (indirectly using an open)

For object create and delete or name change:

Audit event types:

32, 41, 42, 43, 44, 45, 47, 48, 53, 54, 55, 64

RACF callable service:

ck_owner_two_files, ckpriv, makeFSP, R_audit

z/OS UNIX services:

chdir, chmount, chmountsetuid, link, mkdir, mknod, mount, mountsetuid, open(new file), remove, rename, rmdir, symlink, unlink, unmount, unmountsetu, vlink, vmkdir, vcreate, vremove, vremovedir, vrename, vsymlink

Note: Chdir, symlink, and vsymlink are included to make it possible to re-create from the audit records the full path name you are using when accessing files. Services other than those listed above are audited with audit event type 42 or 43.

FSSEC

Controls auditing for changes to the security data (FSP and ACL) for file system objects:

Audit event types:

31, 33, 34, 35, 75, 76, 77

RACF callable services:

R_chaudit, R_chmod, R_chown, clear_setid, R_setfacl, R_setfsecl

z/OS UNIX services:

chaudit, chmod, chown, fchaudit, fchmod, fchown, write, chattr, fchattr, setfacl, vsetattr, vreadwrite

Note: Event type 75, SETFACL, has a separate audit record created for each ACL entry which is added, modified, or deleted.

IPCOBJ

Specifies auditing options for IPC accesses. For access control and for z/OS UNIX user identifier (UID), z/OS UNIX group identifier (GID), and mode changes, use SETROPTS LOGOPTIONS. For object create and delete, use SETROPTS AUDIT (see note below).

For access control or UID, GID, or mode changes:

Audit event types:

60, 62

RACF callable services:

ck_IPC_access, R_IPC_ctl

z/OS UNIX services:

msgctl, msgget, msgsnd, msgrcv, semctl, semget, semop, shmat, shmctl, shmget, w_getipc

For object create and delete or for remove ID:

Audit event types:

61, 62

RACF callable services:

makeISP, R_IPC_ctl

z/OS UNIX services:

msgctl, msgget, semctl, semget, shmctl, shmget

PROCESS

Controls auditing of changes to the UIDs and GIDs of processes and changing of the Osigset action, thread limit, and other privileged operations using the SETROPTS LOGOPTIONS, and controls auditing of dubbing, undubbing, and server registration of processes using SETROPTS AUDIT (see note below).

For UID/GID, Osigset and thread limit changes, and other privileged operations:

Audit event types:

36, 49, 50, 51, 52, 57, 63

RACF callable services:

R_exec, R_setuid, R_setgid, R_seteuid, R_setegid, ck_priv

z/OS UNIX services:

_console, exec, __login, server_init, setuid, setgid, seteuid, setegid, shutdown_reg, sigaction, spawn, swap services, thlmt, WLMC

For process dubbing, undubbing, and registration:

Audit event types:

38, 39, 57

Note: Unsuccessful process dubs (38 events) are always audited.

RACF callable services:

initUSP, delete_USP, ck_priv

z/OS UNIX services:

first syscall for a process, dub, _exit, undub, vregister

PROCACT

Controls auditing of functions that look at data from or effect other processes:

Audit event types:

37, 40, 46, 58, 65

RACF callable services:

ck_process_owner, R_ptrace

z/OS UNIX services:

getpsent, kill, ptrace, recv, recvmsg, sendmsg

Audit records are written for getpsent only during the following configuration: SETROPTS LOGOPTIONS (ALWAYS).

Note about using SETROPTS AUDIT: For the services listed whose auditing is controlled by SETROPTS AUDIT, all successful requests are audited. Failures for these services are audited by the authority check that actually failed (for example, an access check to a FACILITY class profile, or an access check controlled by the FSOBJ or DIRACC classes). To audit these, use LOGOPTIONS(FAILURES) for the appropriate classes.

Auditable events

RACF writes audit records for the z/OS UNIX System Services auditable events in SMF type 80 records. File owners and auditors can establish separate sets of auditing rules, and can also specify auditing for each file and directory. For more information about these event codes, see *z/OS Security Server RACF Macros and Interfaces*.

Commands

You can control auditing by using the existing SETROPTS LOGOPTIONS and SETROPTS AUDIT.

Use SETROPTS LOGOPTIONS to specify logging options for all the classes associated with z/OS UNIX System Services:

- DIRSRCH: Directory searches
- DIRACC: Access checks for read/write accesses to directories
- FSOBJ: Access checks for files and directories
- FSSEC: Changes to file system security
- IPCOBJ: Access checks for objects and changes to UIDs, GIDs, and modes
- PROCESS: Changes to UIDs and GIDs of processes and to privileged operations requiring superuser authority
- PROCACT: Functions that look at data from other processes or effect other processes

Here is an example:

```
SETROPTS LOGOPTIONS(FAILURES(DIRSRCH,DIRACC))
```

In addition, you can use the SETROPTS AUDIT option to control auditing for the FSOBJ, IPCOBJ, and the PROCESS classes.

- FSOBJ: Successful creation and deletion of file system objects
- IPCOBJ: Successful creation and deletion of objects (message queues, semaphores, and shared memory segments)
- PROCESS: Successful dubbing or undubbing of a process

Here is an example:

```
SETROPTS AUDIT(FSOBJ,PROCESS)
```

Audit options for file and directory levels

The following audit options for file and directory levels are stored inside the HFS along with the *file permission bits*:

- don't_audit
- audit_access_allowed
- audit_access_failed
- audit_all_access

A directory is just a special-purpose file. When a file or a directory is created, default audit options are assigned. Different defaults are set for users and auditors. The same audit option is used no matter what kind of access is attempted (read, write, or execute).

When a file is created, these are the default audit options:

- User audit options: for all access types, audit_access_failed
- Auditor audit options: for all access types, don't_audit

To change the audit options, you must use **chaudit**, a z/OS UNIX System Services Shell and Utilities feature. For complete information about this command, see *z/OS UNIX System Services Command Reference*. There are restrictions on who can change these options.

- For user audit options, you must be the owner of the file.
- For auditor audit options, you must have the RACF AUDITOR attribute. You can then change the auditor audit options for any file in the file system.

The default file-level audit options control the auditing of directory and file accesses. These defaults are only used for a particular class (DIRSRCH, DIRACC, or FSOBJ) if SETROPTS LOGOPTIONS(DEFAULT(*class*)) has been issued for that class.

Auditing for superuser authority in the UNIXPRIV class

If you use profiles in the UNIXPRIV class to control superuser authorities, you can use the same profiles for auditing.

UNIXPRIV

Controls auditing of superuser authorities:

Audit event type:

2

RACF callable services:

ck_access, ck_owner_two_files, ck_priv, ck_process_owner,
R_chown, R_IPC_ctl, R_ptrace, R_chmod

z/OS UNIX services:

chmod, chmount, chmountsetuid, chown, getpsent, kill, link,
mkdir, mount, mountsetuid, nice, open, opendir, pfsctl, ptrace,
quiesce, quiescesetu, readlink, realpath, rename, rmdir, setpriority,
shmmcv, stat, symlink, unlink, unmount, unmountsetu, unquiesce,
unquiescesu, vregister

RACF logs successful attempts to use superuser authorities. If you want to check the use of superuser authority for specific resources, you can audit successful uses of the UNIXPRIV profiles. RACF logs failed attempts to use SHARED.IDS in the UNIXPRIV class. For other UNIXPRIV resources, no audit record is written to show authorization failures in the UNIXPRIV class.

For example, to audit the successful uses of the `kill()` function, granted by the `SUPERUSER.PROCESS.KILL` profile, set the audit options as follows:

```
RALTER UNIXPRIV SUPERUSER.PROCESS.KILL AUDIT(SUCCESS(READ))
```

LOG=NOFAIL is specified on all authorization checks in the UNIXPRIV class, except for SHARED.IDS. Therefore, RACF does not log failures, even when you specify AUDIT(FAILURES) or AUDIT(ALL) in the profile. RACF also ignores any SETROPTS LOGOPTIONS settings in the UNIXPRIV class because the RACROUTE REQUEST=FASTAUTH request performs all authorization checks in that class.

It is possible to see multiple audit records for the same operation, as described in the following example:

1. You are auditing successful uses of the SUPERUSER.PROCESS.KILL profile.
2. You also issued the SETROPTS LOGOPTIONS(SUCSESSES(PROCACT)) command to audit success in the PROCACT class.

Note: This is not recommended because of the large number of audit records it could produce.

3. User LAURIE has UID 40 and READ access to the SUPERUSER.PROCESS.KILL profile in the UNIXPRIV class.
4. User LAURIE issued the kill() function for another user's process.

The kill() function succeeds and RACF writes two audit records as a result of:

- Auditing for the PROCACT class
- A RACROUTE REQUEST=FASTAUTH call in the UNIXPRIV class

For more information about the UNIXPRIV class, see *z/OS Security Server RACF Security Administrator's Guide*.

Auditing for the RACF remote sharing facility (RRSF)

The RACF remote sharing facility (RRSF) allows you to administer and maintain RACF databases that are distributed throughout the enterprise. It helps to ensure that data integrity is kept across system or network failures and delays. You are informed when key events occur and returns output to view at your convenience.

RRSF uses the RACF subsystem address space. The address space supports a library that contains information that is needed by the remote sharing facility. For more information, see *z/OS Security Server RACF Security Administrator's Guide* and *z/OS Security Server RACF System Programmer's Guide*.

RACF MVS operator commands for RRSF

The following RACF MVS operator commands can be used to control the RRSF function:

- **SET**

A RACF command that establishes operational characteristics for RRSF. It also has a list capability that provides a summary of the information related to the RRSF node that the command runs on.

- **TARGET**

A RACF command that defines, to the logical node being configured, the communication attributes and associated information for RRSF nodes with which it can potentially communicate. It also has a list capability that provides a list of attributes associated with each of the target nodes defined to the RRSF node.

- **STOP**

A RACF command that stops the RACF subsystem address space without losing any requests that may be waiting for completion.

- **RESTART**

A RACF command that restarts RRSF subtasks in the RACF subsystem address space.

For more information about these commands, including issuing options, see *z/OS Security Server RACF Command Language Reference*.

The SET and TARGET commands generate SMF records, depending on where these commands are issued. If these commands are issued from the operator console and the auditing attributes have been defined in the OPERCMDS profile that covers the command, auditing takes place accordingly. If, however, the SET and TARGET commands are issued from the RACF parameter library, either because of RRSF initialization during RACF subsystem address space initialization or because of a SET command with the INCLUDE keyword specified, authorization to individual commands is not checked and no auditing occurs.

RACF also writes audit records for the STOP and RESTART commands, if auditing attributes have been defined in the OPERCMDS profile for the command.

As with all operator commands protected by an OPERCMDS profile, if an SMF record is cut, the command-up to 255 characters-is included in the audit record.

Directed commands for RRSF

The AT and ONLYAT keywords are used to direct specific RACF commands to run in the RACF subsystem address space on the specified local or remote RRSF node, under the authorization of the specified user ID.

- **AT**

A keyword used to direct specific RACF commands to run at a specified target node in the RACF subsystem address space under the authority of a specified user ID. When using the AT keyword, automatic command direction can occur from the target node. For example, if the RACF command was sent from node A to node B and automatic command direction is enabled between node B and node C, the command also takes effect on node C.

The AT keyword is controlled by profiles in the RRSFDATA class. In order for an authorization check to be audited, the audit attributes must be set in the general resource profile in the RRSFDATA class. In this case, the SMF record is generated on the issuing node. If the profile does not exist, the command fails and no auditing occurs.

Audit records produced from the authorization check include a LOGSTR relocate section that contains a copy of the command image. The LOGSTR is limited to a length of 255 characters. Longer strings may be truncated. The following command keywords are suppressed in the LOGSTR data:

Command	Keyword(Field)
ADDUSER	PASSWORD() PHRASE()
ALTUSER	PASSWORD() PHRASE()

Command	Keyword(Field)
PASSWORD	PASSWORD()
RDEFINE	SESSION(SESSKEY()) SSIGNON(KEYMASKED()) SSIGNON(KEYENCRYPTED())
SETROPTS	RVARYPW(SWITCH()) RVARYPW(STATUS())

For information about the RRSFDATA class, see *z/OS Security Server RACF Security Administrator's Guide*.

- **ONLYAT**

A keyword used to fix an “out of sync” condition between RACF databases in the RRSF network. It works much like the AT keyword, but automatic command direction does not occur on the target node. No auditing occurs on the issuing node from failures or successes, regardless of audit settings such as SETROPTS SAUDIT or ALTUSER UAUDIT. However, auditing still occurs on the target system. To use the ONLYAT keyword, you must have the SPECIAL attribute.

For more information about these commands, refer to *z/OS Security Server RACF Command Language Reference*.

After the command has passed authorization checking for command direction, the command is sent to the target system. Records that are cut on the target system automatically include, in the audit record, information to reflect command direction. The issuing node and user ID is added to the SMF record by including a type 44 relocate section whenever the command is directed. Since this relocate section is generally used for segments, a dummy segment name of CMDSRC (command source) is used. The type 44 relocate section format is as follows:

Byte	Description	Value
Byte 1	Bit string	Not used
Byte 2-9	Name of segment	CMDSRC
Byte 10	Length of subkeyword	15
Byte 11-25	Subkeyword	ORIGINATED_FROM
Variable length	Origination data	node.userid,DIRECTED BY_AT/BY_ONLYAT/ AUTOMATICALLY

This additional information is added to the audit record that gets generated on the target system. Auditing is based on the same criteria as if the command had been issued on the target system. No command auditing takes place on the issuing system.

If commands run in the RACF subsystem address space, the SMF80JBN is filled in with the subsystem name. SMF80UID contains the user ID under whose authority the command is running.

Automatically directed commands for RRSF

Automatic command direction is designed primarily to ensure that RACF profiles remain synchronized between two or more remote nodes regarding RACF TSO commands. It is similar to command direction in the following ways:

- Like directed commands using the AT keyword, automatically directed commands are controlled by profiles in the RRSFDATA class. An audit record produced from this authority check contains a LOGSTR relocate section.
- Like command direction, the command is sent to the target node after it passes authorization checking. All records produced on the target node contain a type 44 relocate section, which identifies the source of the command.

See “Directed commands for RRSF” on page 39 for detailed information about the LOGSTR and type 44 relocate sections. For information about the RRSFDATA class, see *z/OS Security Server RACF Security Administrator’s Guide*.

Automatically directed application updates for RRSF

Automatic direction of application updates is designed primarily to ensure that RACF profiles remain synchronized between two or more nodes regarding application updates. It is similar to command direction in the following ways:

- Like directed commands using the AT keyword, automatically directed application updates are controlled by profiles in the RRSFDATA class. An audit record produced from this authority check contains a LOGSTR relocate section. This authority check is only done on the originating node before propagation. Any corresponding SMF records are only created on the originating node.

On the RACROUTE REQUEST=AUTH application update for the RRSFDATA class, the LOGSTR parameter is specified to describe the application update. The LOGSTR parameter contains:

```
request FOR CLASS class-name, PROFILE profile-name:
APPLICATION UPDATE PROPAGATION ATTEMPTED
```

Since the maximum LOGSTR length is 255, this information may be truncated if the profile name is very long. The possible values of *request* are:

- RACROUTE REQUEST=DEFINE TYPE=CHGVOL
- RACROUTE REQUEST=DEFINE TYPE=ADDVOL
- RACROUTE REQUEST=DEFINE TYPE=DELETE
- RACROUTE REQUEST=DEFINE TYPE=DEFINE
- RACROUTE REQUEST=DEFINE TYPE=DEFINE,NEWNAME
- RACROUTE REQUEST=EXTRACT
- ICHEINTY ALTER OPERATION
- ICHEINTY ADD OPERATION
- ICHEINTY DELETE OPERATION
- ICHEINTY DELETEA OPERATION
- ICHEINTY RENAME OPERATION

For information about the RRSFDATA class, see *z/OS Security Server RACF Security Administrator’s Guide*.

- Like command direction, the application update is sent to the target node after it passes authorization checking. No auditing is done for RACROUTE REQUEST=EXTRACT and ICHEINTY requests on the system on which they run. Auditing is done for RACROUTE REQUEST=DEFINE requests. Records that are cut on the target system automatically include, in the audit record, information to reflect an automatically directed application update. The issuing node and user ID is added to the SMF record by including a type 44 relocate section whenever a RACROUTE REQUEST=DEFINE application update has been

automatically directed. Since this relocate section is generally used for segments, a dummy segment name of APPLSRC (application source) is used. The type 44 relocate section format is as follows:

Byte	Description	Value
Byte 1	Bit string	Not used
Byte 2-9	Name of segment	APPLSRC
Byte 10	Length of subkeyword	15
Byte 11-25	Subkeyword	ORIGINATED_FROM
Variable length	Origination data	node.userid,DIRECTED_AUTOMATICALLY

This additional information is added to the audit record that gets generated on the target system. Auditing is based on the same criteria as if the application update had been issued on the target system.

Automatically directed passwords for RRSF

Automatic password direction is designed primarily to ensure that RACF user profiles remain synchronized between two or more remote nodes regarding RACF passwords. There is no need to establish RACLINK PEER PWSYNC associations.

Like directed commands using the AT keyword, automatically directed passwords are controlled by profiles in the RRSFDATA class on the issuing node. Audit records contain a LOGSTR relocate section only for password changes resulting from a RACF command and do not contain the new password. Automatically directed passwords are not audited on the target node.

RACF considers password changes resulting from the ADDUSER, ALTUSER, or PASSWORD command to be automatically directed commands and can be audited on the target node like other commands. Password changes resulting from a logon or from RACROUTE or ICHEINTY calls are not audited on the target node.

The RACLINK command

The RACLINK command is a RACF TSO command used to:

- Define, approve, and undefine user ID associations
- Enable password synchronization between pairs of user IDs
- List information related to user ID associations

The DEFINE keyword used with the RACLINK command allows two user IDs to form a user ID association. A user ID association enables RACF users to take advantage of command direction and password synchronization.

Several user IDs are involved in RACLINK processing. These are:

Issuing user ID

is the user who issues the RACLINK command from the source system.

Source user ID

is the user on the source system for whom an association is created.

Target user ID

is the user on the target system for whom an association is created.

Authorization user ID

is the user on the target system used by the authority checking program for processing the RACLINK request. For information about the associated list used to determine the authorization user ID, see *z/OS Security Server RACF Security Administrator's Guide*.

There are three potential phases for RACLINK commands. Each phase uses event code 59, which is described in *z/OS Security Server RACF Macros and Interfaces*. The phases are:

Phase 1: Local issuance of the RACLINK command

In phase 1, auditing is based on the auditing criteria of the issuing user ID. Auditing can be set up for both DEFINE and PWSYNC keywords.

Phase 2: Processing on the target node

In phase 2, auditing is based on the auditing criteria of the authorization user ID. If no user ID in the associated list is found to have the appropriate authorization, auditing is based on the auditing criteria of the target user ID.

Phase 3: Response from the target node

Phase 3 occurs only with the DEFINE keyword. If no error is detected, auditing is based on the issuing user ID. This allows you to determine what took place on the target system without having to log on to the target system to view the SMF records. These audit records get cut on the issuing node based on the auditing criteria of the issuing user ID.

Although phases 2 and 3 are not commands, SMF records are generated as if they were. For phase 2, auditing is based on either the authorization user ID or the target user ID, as if a command had been issued. For phase 3, auditing is based on the issuing user ID.

As with directed commands, the DEFINE and PWSYNC parameters of the RACLINK command are controlled by profiles in the RRSFDATA class. Auditing for this authorization check occurs on the issuing system and is determined by the auditing attributes in the RRSFDATA class profiles. If either the DEFINE or PWSYNC keyword is specified and the corresponding profile does not exist, the command fails and no auditing takes place.

To better understand this process, see the examples that follow.

Example 1

In this example, user JAMES on node NODEA has the SPECIAL attribute and SETROPTS SAUDIT is in effect. On node NODEA, the following RRSFDATA profiles exist, each with UACC(READ) and AUDIT(SUCCESS):

```
RACLINK.DEFINE.NODEB
RACLINK.PWSYNC.NODEB
```

On node NODEB, user JIM has JIMSPW as a password, has the SPECIAL attribute, and SETROPTS SAUDIT is in effect.

- Phase 1

When JAMES issues:

```
RACLINK DEFINE(NODEB.JIM/JIMSPW) PWSYNC
```

1. Two separate audit records are produced (one for the RACLINK DEFINE in general and the second for the PWSYNC). These records are produced

because user ID JAMES passed both authority checks by having UACC(READ) and AUDIT(SUCCESS) set in each of the profiles. The LOGSTR data in each contains a copy of the command image, up to a maximum of 255 characters.

2. Authority checking has passed, so there can also be an SMF record based on the local issuance of RACLINK (event code 59). Because JAMES has the SPECIAL attribute and SETROPTS SAUDIT is in effect, this third audit record is produced with event code qualifier 0.
- Phase 2
On node NODEB the TARGET PROCESSING phase begins with checking the supplied password for JIM (in this case, JIMSPW is correct).
Auditing takes place on the target user ID, JIM, because JIM has the SPECIAL attribute and the target node (NODEB) has SETROPTS SAUDIT in effect. This record, cut for RACLINK (event code 59, qualifier 0), is the fourth audit record to be produced, although it is the first record cut on node NODEB.
 - Phase 3
On node NODEA, a fifth audit record is produced (event code 59, qualifier 0) for the final phase because JAMES has the SPECIAL attribute and SAUDIT is still in effect for the issuing node (NODEA).

Example 2

For this example, assume that Example 1 has completed successfully.

- Phase 1
User JAMES on node NODEA issues:

```
RACLINK ID(MARY) DEFINE(NODEB.MARY) PWSYNC
```

As in phase 1 of example 1, the authorization checking on SPECIAL user JAMES yields two SMF records, and a RACLINK record is produced. A significant difference in the event code 59 record is that while JAMES is still the issuing user ID, MARY is the source user ID.
- Phase 2
In this phase, target processing uses JAMES' association with JIM. This means that although MARY (on NODEB) is the target user ID, JIM is the authorization ID. Therefore, auditing is based on JIM. Because JIM has the SPECIAL attribute and SETROPTS SAUDIT is in effect on NODEB, an event code 59 record (qualifier 0) is produced. This indicates that the association has been established successfully.
- Phase 3
As in example 1, auditing for this phase is based on JAMES. The issuing user ID is JAMES, the source user ID is MARY (from NODEA), the target user ID is MARY (from NODEB), and there exists an authorization user ID, JAMES.
An event code 59 record, which contains this data, is produced because JAMES has the SPECIAL attribute and SAUDIT is still in effect on the issuing node (NODEA).

Checking DB2 authorization

When you use the RACF/DB2 external security module to check authorization, RACF simulates DB2 authorization. Each DB2 SQL statement, command, or utility specifies a particular set of privileges and authorities. The RACF/DB2 external security module checks the RACF profiles that correspond to each set.

Table 1. How RACF Simulates DB2 Authorization Checking

In DB2:	Maps to in RACF:
Object types	Class names
Privileges	Profile names
Authorities	Administrative authority profile names
Privilege sets	Profile checking

Note: When an input ACEE (XAPLACEE) is not provided, the RACF/DB2 external security module returns the authority checking responsibility to DB2. For more information, see *DB2 Administration Guide*. For details about authority checking with the RACF/DB2 external security module, see *z/OS Security Server RACF Security Administrator's Guide*.

Example of profile checking

DB2 privilege sets map to RACF profile checking. This example describes RACF profile checking for the SELECT statement.

When RACF checks authorization, the requester must own the object or have read access to one of the following profiles:

Class Profile Type

MDSNTB

subsystem.table-name.SELECT

(gives access to the table)

DSNADM

subsystem.database-name.DBADM

(gives access to the database that holds the table)

DSNADM

subsystem.SYSCTRL

(bypassed for user tables)

DSNADM

subsystem.SYSADM

RACF produces an SMF record for a failure only after checking the entire list of profiles and the requestor fails to meet any of the requirements. RACF does not produce an audit record if:

- The requestor meets any of the requirements and access is granted
- The RACF/DB2 external security module returns the authority checking responsibility to DB2

An audit record is produced for the first resource that has auditing indicated by the covering profile and receives a return code of 8.

RACF produces an SMF record for a success when the requestor indicates that should be performed.

For a description of the RACF classes, see *z/OS Security Server RACF System Programmer's Guide*.

Using the log string (LOGSTR) data

The log string data consists of information that can help you audit DB2 successfully. DB2 uses the XAPL parameter list (DSNDXAPL macro) to pass the log string information to the RACF/DB2 external security module. The "LOGSTR=" parameter of the RACROUTE REQUEST=FASTAUTH request contains the input portion of XAPL and does the following:

- Identifies the RACF/DB2 external security module request that caused RACF to create the audit record. The RACF profile causing the audit record to be cut could be a profile that provides a DB2 administrative authority and might not identify the specific DB2 resource being accessed. The LOGSTR data contains values from the XAPL parameter list that are necessary to identify that unique request from the RACF/DB2 external security module.
- Links RACF SMF 80 records with DB2 IFCID 314 records. Each invocation of the RACF/DB2 external security module might produce an SMF 80 record. DB2 might produce a DB2 IFCID 314 record in addition to the SMF 80 records cut by RACF. You can determine that the records were cut for the same RACF/DB2 external security module request if the LOGSTR_TIME and LOGSTR_USER values in the SMF 80 record match the XAPLSTCK and XAPLUCHK values in the IFCID 314 request. The RACF/DB2 external security module uses these time and user values created from the LOGSTR data to link the RACF and DB2 information.

The LOGSTR data includes the following ordered information. A blank space separates each field, as indicated in the table.

Table 2. LOGSTR Data

Log string data	Length	XAPL field Name	Description
LOGSTR_DATA	DS 0CL241		
LOGSTR_TIME	DS CL8	XAPLSTCK	Time
	DS CL1		
LOGSTR_USER	DS CL8	XAPLUCHK	User
	DS CL1		
LOGSTR_SUBSYSTEM	DS CL4	XAPLGPAT	Subsystem/DB2 group attach name
	DS CL1		
LOGSTR_OBJTYPE	DS CL1	XAPLTYPE	Object type
	DS CL1		
LOGSTR_FLAGS	DS 0CL16	XAPLFLG1	Flags: The flags in this field are declared as BL1. It is translated to CL16 in the LOGSTR data field, containing one character for each bit and a blank space between each one. <ul style="list-style-type: none"> • If the bit is on, "Y" is inserted • If the bit is off, "N" is inserted • Reserved bits are left blank
LOGSTR_SECNDRY_ID	DS CL1	XAPLFLG1	Secondary ID (Y/N)
	DS CL1		
LOGSTR_USERTAB	DS CL1	XAPLFLG1	User table (Y/N)
	DS CL13		Reserved (blank)
LOGSTR_OBJNAME	DS CL20	XAPLOBJN	Object name
	DS CL1		

Table 2. LOGSTR Data (continued)

Log string data	Length	XAPL field Name	Description
LOGSTR_OBJOWNER	DS CL20	XAPLOWNQ	Object owner or qualifier
	DS CL1		
LOGSTR_REL1	DS CL20	XAPLREL1	Related information 1
	DS CL1		
LOGSTR_REL2	DS CL20	XAPLREL2	Related information 2: This is the first 20 bytes of the XAPLREL2 field.
	DS CL1		
LOGSTR_PRIV	DS CL3	XAPLPRIV	Privilege
	DS CL1		
LOGSTR_SOURCE	DS CL1	XAPLFROM	Source of the request
	DS CL1		
LOGSTR_CLASS	DS CL8		Class name
	DS CL1		
LOGSTR_ENTY	DS CL100		Entity name: This is the first resource checked for a specific request.

Examples for setting audit controls for DB2

The RACF/DB2 external security module attempts to produce an audit record after checking the list of profiles.

Example 1

In this example, user SIVLE wants to use the DB2 SELECT statement to retrieve table CADDY, which is in database DSNDB04, from DB2 subsystem CARS.

1. Does SIVLE own the table?
Because SIVLE does not own the table, the table name qualifier passed from DB2 does not match the user ID. In this case, RACF does not check a profile, so no audit record is written.
2. Does SIVLE have SELECT authority?
RACF checks CARS.CADDY.SELECT in class MDSNTB. SIVLE does not have SELECT authority. If SIVLE does not meet any of the other requirements, this is the "first failing resource."
3. Does SIVLE have database administrator authority?
RACF checks CARS.DSNDB04.DBADM in class DSNADM. SIVLE does not have this authority.
4. Does SIVLE have system administrator authority?
RACF checks CARS.SYSADM in class DSNADM. SIVLE does not have this authority.

Because SIVLE has none of the correct authorities, RACF produces SMF records relating to the first failure it encountered. Although SIVLE did not own the table, no profiles were checked and failures were not audited. Therefore, the first failing resource is CARS.CADDY.SELECT. RACF produces an audit record for this resource and identifies it in message ICH408I. The data is contained in the LOGSTR information and can be used in a report.

Example 2

In this example, user SIVLE issues a START PROCEDURE TUNEUP request for DB2 subsystem CARS.

1. Does SIVLE have SYSOPR authority?
RACF checks CARS.SYSOPR in class DSNADM. SIVLE does not have SYSOPR authority.
2. Does SIVLE have system administrator authority?
RACF checks CARS.SYSADM in class DSNADM. SIVLE does not have this authority.

Because SIVLE has none of the correct authorities, RACF produces SMF records relating to the failure. The failure record is cut for resource CARS.SYSOPR, which was the first failing resource. The LOGSTR information can help you to determine what SIVLE wanted to do. It includes the object type, object name, and privilege, which you can use in a report.

Auditing security events for other components

It is possible to set audit controls for other components using RACF commands, such as EIM. For more details see *z/OS Integrated Security Services EIM Guide and Reference*.

Chapter 3. The RACF SMF data unload utility

RACF audit data is a record of an installation's security relevant events. This data is used to verify the effectiveness of an installation's security policy, determine whether the installation's security objectives are being met, and identify unexpected security relevant events.

The RACF SMF data unload utility (IRRADU00) enables installations to create a sequential file from the security relevant audit data. The sequential file can be used in several ways: viewed directly, used as input for installation-written programs, manipulated with sort/merge utilities, output to an XML-formatted file for viewing on a web browser, or uploaded to a database manager (for example, DB2) to process complex inquiries and create installation-tailored reports. It is not intended to be used directly as input to RACF commands.

Operational considerations

IRRADU00 processes these types of SMF records:

Type 30

Job initiation - Subtype 1 (Job initiation) and subtype 5 (Job termination)

Type 80

Resource access - No subtypes in record

Type 81

RACF initialization - No subtypes in record

Type 83

- Subtype 1, Data sets affected by a security label change
- Subtype 2, EIM
- Subtype 3, LDAP
- Subtype 4, Remote audit
- Subtype 5, Websphere
- Subtype 6, TKLM

To correlate the RACF audit data with the data unloaded by IRRADU00, see the description of the SMF records contained in *z/OS Security Server RACF Macros and Interfaces*. For more details about working with subtype 3 LDAP audit records, see *z/OS IBM Tivoli Directory Server Administration and Use for z/OS*.

Using IRRADU00

The RACF SMF data unload utility uses the SMF Dump Utilities (IFASMFDP or IFASMFDL) as the “driver” module to control its invocation. The RACF SMF data unload utility is invoked as USER2 and USER3 exits to IFASMFDP or IFASMFDL. To request RACF SMF data unload utility processing, enter the names of the RACF SMF data unload utility modules (IRRADU00 and IRRADU86) in the SYSIN data stream for IFASMFDP or IFASMFDL.

The following job control statements are necessary for executing IRRADU00:

JOB Initiates the job.

EXEC Specifies the program name (PGM=IFASMFDP or PGM=IFASMF DL) or, if the job control statements are in a procedure library, the procedure name.

SYSPRINT DD

Defines a sequential message data set for the messages produced by IFASMFDP or IFASMF DL.

SYSIN DD

Defines a sequential input data set for the SMF Dump Utility control statements. These statements must include the USER2 (IRRADU00) and USER3 (IRRADU86) statements for invoking the RACF SMF data unload utility. Additional IFASMFDP or IFASMF DL control statements can be used to select records based on date, time, and SMF system ID. IFASMFDP and IFASMF DL use defaults for control parameters that are allowed to default. In particular, the default value for the OPTIONS parameter is ALL, which causes the input data set to be reset so that it can be reused. See *z/OS MVS System Management Facilities (SMF)* for information on overriding defaults.

Note: IRRADU00 is called once for each IFASMFDP OUTDD or IFASMF DL OUTDD control statement that meets the specified selection criteria. This might result in multiple calls for the same SMF record, causing that record to appear more than once in the SMF unload output.

ADUPRINT DD

Defines a sequential message data set for the messages produced by the RACF SMF data unload utility.

DUMPIN DD

Defines the input SMF data stream.

Note: The ddname DUMPIN can be changed by the control statements that are contained in the SYSIN data stream.

DUMPOUT DD

Defines the output SMF data stream. After the RACF SMF data unload utility processes a record, it returns control to IFASMFDP or IFASMF DL and tells these utilities to continue its processing of the record. This causes IFASMFDP or IFASMF DL to write the record to DUMPOUT. If you do not want to retain these records, allocate DUMPOUT to DUMMY.

Note: The ddname DUMPOUT can be changed by the control statements that are contained in the SYSIN data stream.

OUTDD DD

Defines the single sequential output data set. The output of IRRADU00 is a set of variable length records. This data set must be allocated as a variable length data set, with a logical record length (LRECL) of at least 12288. If a shorter LRECL is supplied, IRRADU00 changes the LRECL to 12288.

IRRADU00 also changes the block size of the data set to be at least four more than the LRECL, unless the block size was set to zero to allow the system to choose the best block size.

XMLFORM DD

Creates an easily readable form of the report in XML. Each data tag appears on its own line. This file can be easily read in any editor, including displayed in a web browser. This data set must be allocated as a variable length data set, with a logical record length (LRECL) of at least 12288. If a shorter LRECL is supplied, IRRADU00 changes the LRECL to 12288.

IRRADU00 also changes the block size of the data set to be at least four more than the LRECL, unless the block size was set to zero to allow the system to choose the best block size.

XMLOUT DD

Creates a compressed form of the report in XML. The resulting XML file will be small in filesize but will not be as easily readable as the output obtained by specifying XMLFORM DD. This data set must be allocated as a variable length data set, with a logical record length (LRECL) of at least 12288. If a shorter LRECL is supplied, IRRADU00 changes the LRECL to 12288. IRRADU00 also changes the block size of the data set to be at least four more than the LRECL, unless the block size was set to zero to allow the system to choose the best block size.

The output type generated is dependent on the statements incorporated in the JCL. Whichever output type is coded in the JCL is the only type that will be generated, according to the following rules:

- Is XMLFORM DD specified in the JCL? If so, that is the only output written
- If that is not specified, is XMLOUT DD specified in the JCL? If so, that is the only output written
- If that is not specified, is OUTDD DD specified in the JCL? If so, that is the only output written
- If none of these types are specified in the JCL, the utility issues message IRR67522I **Open failed for OUTDD**

If more than one DDname is placed in the JCL, the above order (XMLFORM, XMLOUT, OUTDD) is used to see which one is created. The actual order of DD statements in the JCL is irrelevant.

After the RACF SMF data unload utility has processed a record, control is returned to IFASMFDP or IFASMF DL, which writes the record to the ddname that was specified in the IFASMFDP SYSIN or IFASMF DL SYSIN control statement.

Writing your own application

When writing an application to process the output of IRRADU00, you must remember that the output of IRRADU00 can change with new releases of z/OS or when service is applied. Your application should be designed to tolerate these compatible changes in the IRRADU00 output. For example:

- Specific record types may grow in size as new fields are added. These fields are added to the end of the record.
- If the updated records exceed the existing blocksize of either the XML-formatted output (from DDNAME XMLFORM) or the non-XML-formatted output (from DDNAME OUTDD), then IRRADU00 automatically updates the blocksize of the output dataset to the new minimum acceptable blocksize. Message IRR6541I is issued. The IRRADU00/IFASMFDP utility return code is zero.
- New values may be added to existing fields in records.

IRRADU00 example

Figure 3 on page 52 shows an example of JCL to execute the RACF SMF data unload utility. The SMF dump utility (IFASMFDP or IFASMF DL) is used to select records based on the date, time, and SMF system identifier.

Due to restrictions of the SMF dump utilities, IRRADU00 and IRRADU86 must reside in an APF-authorized library. For more information on the SMF dump

utilities, see *z/OS MVS System Management Facilities (SMF)*.

```

//SMFUNLD JOB  , 'SMF DATA UNLOAD',
//          MSGLEVEL=(1,1)
//SMFDUMP EXEC PGM=IFASMFDP
//SYSPRINT DD  SYSOUT=A
//ADUPRINT DD  SYSOUT=A
//OUTDD    DD  DISP=SHR,DSN=USER01.RACF.IRRADU00
//SMFDATA  DD  DISP=SHR,DSN=USER01.RACF.SMFDATA
//SMFOUT   DD  DUMMY
//SYSIN    DD  *
              INDD(SMFDATA,OPTIONS(DUMP))
              OUTDD(SMFOUT,TYPE(000:255))
              ABEND(NORETRY)
              USER2(IRRADU00)
              USER3(IRRADU86)
/*

```

Figure 3. JCL to Invoke the RACF SMF data unload utility

Attention:

Do not confuse SMF utility's use of parameter that is called OUTDD for the normal SMF output of IFASMFDP and RACF's (architecturally fixed) DDNAME called OUTDD.

Return codes for IRRADU00

Table 3. Return codes for the SMF data unload utility (IRRADU00)

Hex (decimal)	Explanation
0 (0)	The SMF dump was successful; no errors were encountered. However, IRRADU00 might not unload some records for one or more of the following reasons: <ul style="list-style-type: none"> • Unexpected event code. The numeric value of the event code is unloaded when the event code is unknown. No message is issued. • Unexpected relocate section in record. • Inability to convert data due to unexpected values. • A pre-RACF 1.9 record was encountered. • The SMF record type 83 subtype is unknown.
4 (4)	The SMF dump was successful; one or more errors were encountered but processing continues. IRRADU00 might not unload some records for one or more of the following reasons: <ul style="list-style-type: none"> • Cannot open ADUPRINT. • Open failed for specified DDNAME. • ABEND during utility processing. • Unable to establish recovery environment. • RACF is not enabled.
8 (8)	The SMF dump was not successful; an error terminated processing.

IRRADU00 output

The RACF SMF data unload utility processes every type 80, 81, and 83 SMF record and every job initiation type 30 record that is selected for processing by the SMF dump utilities, IFASMFDP or IFASMF DL. All of the records are written to a sequential file. For details about the format and content of the records created, see *z/OS Security Server RACF Macros and Interfaces*.

Sample DB2 statements, data definition language statements, load utility statements, and queries are provided in SYS1.SAMPLIB. The members are named IRRADULD, IRRADUTB, and IRRADUQR.

Sample DFSORT and ICETOOL statements are provided in SYS1.SAMPLIB. The member is named IRRICE.

XML samples can be obtained online at the RACF website. For more details, see “Internet sources” on page x and “Using the RACF SMF data unload utility to generate XML documents” on page 65.

Using output from the RACF SMF data unload utility

The output file from the RACF SMF data unload utility can be:

- Viewed directly
- Used as input to your own programs
- Manipulated with sort/merge utilities
- Used as input to a database management system so you can produce reports tailored to your requirements
- Viewed using a web browser

Note: The output file is not intended to be used directly as input to RACF commands.

For audit records created for RACF commands, the exact order and format of the unloaded keywords and operands from the commands are not part of the programming interface. They are contained in fields with names ending in `_SPECIFIED`, `_IGNORED`, and `_FAILED`.

Sort/Merge programs

The RACF SMF data unload utility processes all of the type 80, 81, and 83 SMF records, and all of the job initiation type 30 records that are in the input data stream. If you want a subset of the output records, you can use a standard utility such as DFSORT to select them. For example, the following DFSORT control statements select all the job initiation records. All other record types are excluded.

```
SORT FIELDS=(5,8,CH,A)
INCLUDE COND=(5,8,CH,EQ,C'JOBINIT ')
OPTION VLSHRT
```

For more information about using the DFSORT ICETOOL with the RACF SMF data unload utility, see “Using the DFSORT ICETOOL to create reports” on page 54.

Relational databases

You can use the power of a relational database management system (DBMS), such as DB2, to process the RACF SMF data records. Refer to the following section for details.

XML

RACF SMF data records can be output as XML and then viewed using a web browser. This can give you a better view of the data and use colors to differentiate information. For more details, see “Using the RACF SMF data unload utility to generate XML documents” on page 65.

Using the DFSORT ICETOOL to create reports

IBM's DFSORT product provides a reporting facility called ICETOOL.

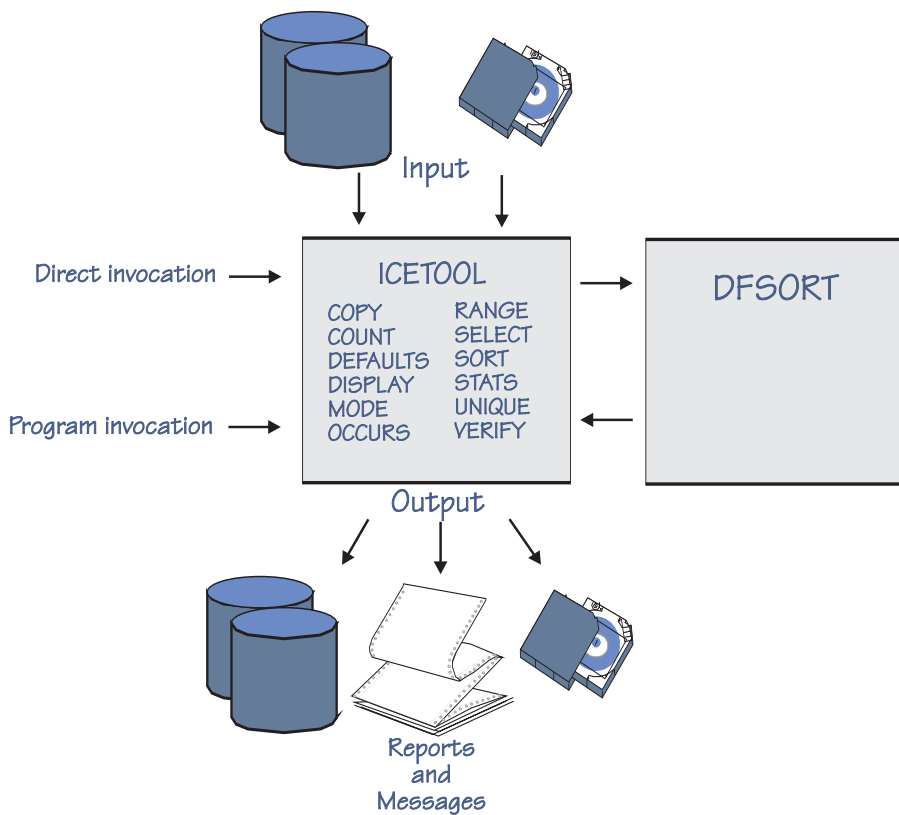


Figure 4. DFSORT's ICETOOL Utility

RACF makes it easy for you to create ICETOOL reports by using the RACFICE PROC, a procedure contained in the IRRICE member of SYS1.SAMPLIB. IRRICE uses DFSORT statements for the selection criteria and ICETOOL statements for the report format for all the reports. The IEBUPDTE utility processes the IRRICE member and creates a data set that contains the report formats and record selection criteria.

Each report consists of these two members of this PDS:

- The report format
- The record selection criteria

The report format

The report format has a 1-4 character name (for example, SELU) that is a member name in the partitioned data set created by the IEBUPDTE utility. The ICETOOL statements control the report format and record summary information, such as SORT, COPY, DISPLAY, and OCCURS statements. An example of a report format member is shown in Figure 5 on page 55. This is the report format member SELU, which is the report format for the "Selected User" report. See *z/OS DFSORT Application Programming Guide* for the complete details of the DFSORT statements.

```

*****
* Name: SELU *
* *
* Find all of the records which are applicable to a specific *
* user ID. *
* *
*****
COPY FROM(ADUDATA) TO(TEMP0001) USING(RACF)
DISPLAY FROM(TEMP0001) LIST(PRINT) -
PAGE -
TITLE('SELU: Events Associated with a Specific User')-
DATE(YMD/) -
TIME(12:) -
BLANK -
ON(63,8,CH) HEADER('User ID') -
ON(72,8,CH) HEADER('Group') -
ON(5,8,CH) HEADER('Event') -
ON(12,8,CH) HEADER('Qualifier') -
ON(23,8,CH) HEADER('Time') -
ON(32,10,CH) HEADER('Date') -
ON(43,4,CH) HEADER('System') -
ON(175,8,CH) HEADER('Terminal') -
ON(184,8,CH) HEADER('Jobname')

```

Figure 5. Member SELU: Selected User Report report format statements

The record selection criteria

The record selection criteria has a name consisting of the report member name followed by CNTL (e.g. SELUCNTL). Record selection is performed using DFSORT control statements, such as SORT and INCLUDE. An example of a record selection member is shown in Figure 6. This is the report selection member SELUCNTL, which is the selection criteria for the “Selected User” report.

```

INCLUDE COND=(63,8,CH,EQ,C'IBMUSER')
OPTION VLSHRT

```

Figure 6. Member SELUCNTL: Selected User Report record selection statements

You can find a list of IRRADU00 reports in “Reports based on the SMF data unload utility (IRRADU00)” on page 56.

You can find a list of IRRDBU00 reports in *z/OS Security Server RACF Security Administrator’s Guide*.

Using the RACFICE PROC to generate reports

You can invoke the ICETOOL utility with the RACFICE PROC. This procedure, which is contained in the IRRICE member of SYS1.SAMPLIB, simplifies the JCL required to execute reports and contains JCL symbolic variables that represent the input to the RACFICE PROC. These variables are:

DBUDATA

Output of IRRDBU00 that is being used as input to the RACFICE PROC

ADUDATA

Output of IRRADU00 that is being used as input to the RACFICE PROC

REPORT

The name of the report that is being generated. "Creating customized reports" on page 57 describes how you can create your own reports.

You do not need to specify each of these variables every time you execute the RACFICE PROC. For example, if you specify the default IRRDBU00 and IRRADU00 data sets in the RACFICE PROC, you create a report (shown in Figure 7) that lists all of the audit records for a specific user with the JCL:

```
//jobname JOB Job card...
//stepname EXEC RACFICE,REPORT=SELU
```

If the default IRRDBU00 or IRRADU00 data sets are not correct, you can override them. For example, if the IRRDBU00 output is in the data set USER01.TEST.IRRDBU00 and the IRRADU00 output is in the data set USER01.TEST.IRRADU00, you should enter:

```
//jobname JOB Job card...
//          SET ADUDATA=USER01.TEST.IRRADU00
//          SET DBUDATA=USER01.TEST.IRRDBU00
//stepname EXEC RACFICE,REPORT=SELU
```

- 1 -		Events Associated with a Specific User			99/02/14	05:14:49 pm	
User ID	Event	Qualifier	Time	Date	System	Terminal	Jobname
IBMUSER	ACCESS	SUCCESS	15:49:59	1999-02-08	IM13	LOCALC10	IBMUSER
IBMUSER	ACCESS	SUCCESS	15:50:27	1999-02-08	IM13	LOCALC10	IBMUSER
IBMUSER	ACCESS	SUCCESS	15:50:27	1999-02-08	IM13	LOCALC10	IBMUSER
IBMUSER	ACCESS	SUCCESS	15:50:28	1999-02-08	IM13	LOCALC10	IBMUSER
IBMUSER	ACCESS	SUCCESS	15:50:29	1999-02-08	IM13	LOCALC10	IBMUSER
IBMUSER	ACCESS	SUCCESS	16:42:11	1999-02-08	IM13		IBMUSERM
IBMUSER	ACCESS	SUCCESS	17:42:36	1999-02-10	IM13	LOCALC10	IBMUSER
IBMUSER	ACCESS	SUCCESS	17:43:11	1999-02-10	IM13	LOCALC10	IBMUSER

Note: This is a sample report only. IBM suggests that you revoke IBMUSER, which would mean that no events would be recorded for this user.

Figure 7. Report for all IRRADU00 Records Associated with a Specific User ID

Reports based on the SMF data unload utility (IRRADU00)

The following reports are based on the output of IRRADU00. You can find a sample of each report in SYS1.SAMPLIB.

Table 4. ICETOOL Reports from IRRADU00 Output

Name	Description	Value
ACD\$	Users who are using automatic command direction	Identifies users who are using the RACF remote sharing facility for automatic command direction
CADU	Count of the IRRADU00 records	Shows the number of SMF-recorded events
CCMD	Count of commands issued (by user)	Shows the command activity for a specific user
ECD\$	Users who are directing commands explicitly	Identifies users who are using the RACF remote sharing facility to explicitly direct commands by specifying "AT(node.user_ID)"
LOGB	Users who log on with LOGON BY, a VM facility	Identifies users who are logging on as other users

Table 4. ICETOOL Reports from IRRADU00 Output (continued)

Name	Description	Value
LOGF	Users with excessive incorrect passwords	Identifies users who have exceeded a “bad password” threshold. This threshold is independent of the SETROPTS PASSWORD(REVOKE(nn)) value
OPER	Accesses allowed because the user has OPERATIONS	Identifies users with the OPERATIONS attribute
PWD\$	Users who are using password synchronization	Identifies users who are using the RACF remote sharing facility
RACL	RACLINK audit records	Identifies users who are using the RACF remote sharing facility
RINC	RACF class initialization information	Shows the status of RACF classes at RACF initialization
SELU	All audit records for a specific user	Reports on all audited events for a user
SPEC	Accesses allowed because the user has SPECIAL	Identifies users with the SPECIAL attribute
TRMF	Excessive incorrect passwords from terminals	Identifies intruders who are attempting to guess passwords but are moving from one ID to another to avoid the revocation of user IDs
VIOL	Access violations	Identifies failed events
WARN	Accesses allowed due to WARNING mode profiles	Identifies events that are allowed but which you might want to prevent in the future

Creating customized reports

You can create your own reports using the RACFICE PROC by following these steps:

1. Identify the records that you want for the report.
 - a. Define the DFSORT statements for the record selection criteria.
 - b. Place them in the RACFICE data set with a unique member name consisting of a 1-4 character report identifier followed by CNTL.

If there is an existing RACFICE report that has similar selection criteria, use it as a model. For example, if you want to report all the access records created when users PATTY, MAXINE, and LAVERNE accessed resources, you need to create DFSORT selection statements that look like Figure 8 and store them in your RACFICE report data set as the PMLCNTL member.

```
INCLUDE COND=(63,8,CH,EQ,C'PATTY',OR,
              63,8,CH,EQ,C'MAXINE',OR,
              63,8,CH,EQ,C'LAVERNE')
OPTION VLSHRT
```

Figure 8. Customized Record Selection Criteria.

Note the similarity of this record selection criteria to the “Selected User” record selection criteria shown in Figure 6 on page 55.

For complete details about the DFSORT statements, see *DFSORT Application Programming Guide*.

2. Identify the report format you want to use.
 - a. Define the ICETOOL statements for the report format.
 - b. Place them in the RACFICE data set with a 1-4 character report identifier that you chose.

If there is an existing RACFICE report that has similar report format, use it as a model. For example, if you wanted your report to contain the user ID, job name, date, time, and status of the access you could use the ICETOOL report statements shown in Figure 9, and store them in your RACFICE report data set as the PML member.

```

COPY    FROM(ADUDATA) TO(TEMP0001) USING(RACF)
DISPLAY FROM(TEMP0001) LIST(PRINT) -
        PAGE -
        TITLE('Patty, Maxine, and Laverne's Accesses')-
        DATE(YMD/) -
        TIME(12:) -
        BLANK -
        ON(63,8,CH) HEADER('User ID') -
        ON(5,8,CH)  HEADER('Event') -
        ON(12,8,CH) HEADER('Qualifier') -
        ON(23,8,CH) HEADER('Time') -
        ON(32,10,CH) HEADER('Date') -
        ON(184,8,CH) HEADER('Job Name')

```

Figure 9. Customized Report Format

Note the similarity of this report format to the “Selected User” report format shown in Figure 5 on page 55.

For complete details about the ICETOOL statements, see *DFSORT Application Programming Guide*.

3. Update the report JCL to invoke the RACFICE PROC with the 1-4 character report identifier you chose, as shown in Figure 10.

```

//jobname JOB Job card...
//stepname EXEC RACFICE,REPORT=PML

```

Figure 10. Customized Report JCL

Using the RACF SMF data unload utility output with DB2

The records produced by the RACF SMF data unload utility are designed to be processed by the DB2 load utility or its equivalent. The definition and control statements for a DB2 utilization of the output, all of which are contained in SYS1.SAMPLIB, are as follows:

- Sample data definition language (DDL) statements to define the relational presentation of the audit information and sample DB2 definitions which perform database and index creation. These are in member IRRADUTB.
- Sample control statements for the DB2 load utility that map the output from the RACF SMF data unload utility. These are in member IRRADULD.
- Sample structured query language (SQL) queries that demonstrate useful inquiries that can be made. These are in member IRRADUQR.

For complete information about DB2, see:

- *DB2 for MVS/ESA Administration Guide*
- *DB2 for MVS/ESA SQL Reference*
- *DB2 for MVS/ESA What's New*

Steps for using IRRADU00 output with DB2

To create and manage a DB2 database that contains the output from the RACF SMF data unload utility, you must:

1. Create one or more DB2 databases.
2. Create one or more DB2 table spaces.
3. Create DB2 tables.
4. Load data into the tables.
5. Reorganize the data in the tables (optional).
6. Create performance statistics (optional).
7. Delete table data (optional).

The first three steps are the initial setup of the database. These steps are required only once, when you first initialize the data manager. After the tables are established, you can import into the DB2 database continuously over a period of time. For example, you can load several days' worth of data before you decide to run the report. This is left up to the installation. At any time, you can delete your current table data. In this case, you reload and reorganize your tables. Also, by using DB2, you can create any required performance statistics at any time.

These steps are similar to those performed when using the RACF database unload utility (IRRDBU00), which is described in *z/OS Security Server RACF Security Administrator's Guide*. Information about the contents of the records produced by RACF SMF data unload utility is in *z/OS Security Server RACF Macros and Interfaces*.

The following sections show examples of the DB2 utility input for each of the steps listed above.

Creating a DB2 database for unloaded RACF SMF data

A DB2 database names a collection of table spaces. The following SQL statement creates a DB2 database for the output of the RACF SMF data unload utility:

```
CREATE DATABASE databasename
```

where *databasename* is supplied by the user.

Creating a DB2 table space

A table space is one or more data sets in which one or more tables are stored. Figure 11 contains examples of SQL statements that create a table space. There are other methods of allocating a table space. For details, see the DB2 documentation.

```
CREATE TABLESPACE tablespacename IN databasename
  LOCKSIZE TABLESPACE
  SEGSIZE 4
  PCTFREE 0
  BUFFERPOOL BP32K
  USING STOGROUP storagegroup
  PRIQTY 20000
  SECQTY 500
  CLOSE NO
  ;
```

Figure 11. Sample SQL Utility Statements Defining a Table Space

The user must supply the name of the table space (*tablespacename*) and the storage group (*storagegroup*). The sample shows a value of 4 for SEGSIZE, 20000 for PRIQTY, and 500 for SECQTY.

Member IRRADUTB in SYS1.SAMPLIB contains statements that create a table space. The sample in IRRADUTB puts all of the tables into one table space. The sample also suggests using a segment size, because segmented table spaces improve performance. You may want to define your own table spaces rather than use table spaces that are defined by the storage group.

You have a number of other options, such as the number of table spaces to use, the type of spaces, and the security for the data. You may want to keep the number of tables per table space fairly small for better performance, and you may want to consider putting the larger tables into separate table spaces.

Creating the DB2 tables

After the database and the table space are created, SQL statements that define the tables are executed. Figure 11 on page 59 contains an example of the SQL statements that are required to create a table for the job initiation records created by the RACF SMF data unload utility.

Member IRRADUTB in SYS1.SAMPLIB contains examples that create separate tables for each record type that is produced by the RACF SMF data unload utility. The user must supply the user ID (*userid*).

```

CREATE TABLE userid.JOBINIT (
    INIT_EVENT_TYPE CHAR(8),
    INIT_EVENT_QUAL CHAR(8),
    INIT_TIME_WRITTEN TIME,
    INIT_DATE_WRITTEN DATE,
    INIT_SYSTEM_SMFID CHAR(4),
    INIT_VIOLATION CHAR(1),
    INIT_USER_NDFND CHAR(1),
    INIT_USER_WARNING CHAR(1),
    INIT_EVT_USER_ID CHAR(8),
    INIT_EVT_GRP_ID CHAR(8),
    INIT_AUTH_NORMAL CHAR(1),
    INIT_AUTH_SPECIAL CHAR(1),
    INIT_AUTH_OPER CHAR(1),
    INIT_AUTH_AUDIT CHAR(1),
    INIT_AUTH_EXIT CHAR(1),
    INIT_AUTH_FAILSFT CHAR(1),
    INIT_AUTH_BYPASS CHAR(1),
    INIT_AUTH_TRUSTED CHAR(1),
    INIT_LOG_CLASS CHAR(1),
    INIT_LOG_USER CHAR(1),
    INIT_LOG_SPECIAL CHAR(1),
    .
    .
    .
    INIT_UTK_SECL CHAR(8),
    INIT_UTK_EXECNODE CHAR(8),
    INIT_UTK_SUSER_ID CHAR(8),
    INIT_UTK_SNODE CHAR(8),
    INIT_UTK_SGRP_ID CHAR(8),
    INIT_UTK_SPOE CHAR(8),
    INIT_UTK_SPCLASS CHAR(8),
    INIT_UTK_USER_ID CHAR(8),
    INIT_UTK_GRP_ID CHAR(8),
    INIT_UTK_DFT_GRP CHAR(1),
    INIT_UTK_DFT_SECL CHAR(1),
    INIT_APPC_LINK CHAR(16),
    INIT_UTK_NETW CHAR(8),
    INIT_RES_NAME VARCHAR(255),
    INIT_CLASS CHAR(8),
    INIT_X500_SUBJECT VARCHAR(255),
    INIT_X500_ISSUER VARCHAR(255),
    INIT_SERVSECL CHAR(8),
    INIT_SERV_POENAME CHAR(64),
    INIT_CTX_USER VARCHAR(510),
    INIT_CTX_REG VARCHAR(255),
    INIT_CTX_HOST CHAR(128),
    INIT_CTX_MECH CHAR(16)
) IN dbname.tablespace

```

Figure 12. Sample SQL Utility Statements Creating a Table

Loading the DB2 tables

Figure 13 on page 62 shows the statements that are required to load the job initiation records. The IRRADULD member of SYS1.SAMPLIB contains statements that load all of the record types produced by the RACF SMF data unload utility. You can choose not to load some of the tables.

```

LOAD DATA INDDN ddname RESUME YES LOG NO
INTO TABLE userid.JOBINIT
WHEN(1:8)='JOBINIT' (
  INIT_EVENT_TYPE POSITION(1:8) CHAR(8),
  INIT_EVENT_QUAL POSITION(10:17) CHAR(8),
  INIT_TIME_WRITTEN POSITION(19:26) TIME EXTERNAL(8),
  INIT_DATE_WRITTEN POSITION(28:37) DATE EXTERNAL(10),
  INIT_SYSTEM_SMFID POSITION(39:42) CHAR(4),
  INIT_VIOLATION POSITION(44:44) CHAR(1),
  INIT_USER_NDFND POSITION(49:49) CHAR(1),
  INIT_USER_WARNING POSITION(54:54) CHAR(1),
  INIT_EVT_USER_ID POSITION(59:66) CHAR(8),
  INIT_EVT_GRP_ID POSITION(68:75) CHAR(8),
  .
  .
  .
  INIT_AUTH_OMVSSU POSITION(258:258) CHAR(1),
  INIT_AUTH_OMVSSYS POSITION(263:263) CHAR(1),
  INIT_USR_SECL POSITION(268:275) CHAR(8),
  INIT_RACF_VERSION POSITION(277:280) CHAR(4),
  INIT_APPL POSITION(282:289) CHAR(8),
  INIT_LOGSTR POSITION(291:545) CHAR(255),
  INIT_BAD_JOBNAME POSITION(547:554) CHAR(8),
  INIT_USER_NAME POSITION(556:575) CHAR(20),
  INIT_UTK_ENCR POSITION(577:577) CHAR(1),
  INIT_UTK_PRE19 POSITION(582:582) CHAR(1),
  INIT_UTK_VERPROF POSITION(587:587) CHAR(1),
  INIT_UTK_NJEUNUSR POSITION(592:592) CHAR(1),
  INIT_UTK_LOGUSR POSITION(597:597) CHAR(1),
  INIT_UTK_SPECIAL POSITION(602:602) CHAR(1),
  INIT_UTK_DEFAULT POSITION(607:607) CHAR(1),
  INIT_UTK_UNKNUSR POSITION(612:612) CHAR(1),
  INIT_UTK_ERROR POSITION(617:617) CHAR(1),
  INIT_UTK_TRUSTED POSITION(622:622) CHAR(1),
  INIT_UTK_SESSTYPE POSITION(627:634) CHAR(8),
  INIT_UTK_SURROGAT POSITION(636:636) CHAR(1),
  INIT_UTK_REMOTE POSITION(641:641) CHAR(1),
  INIT_UTK_PRIV POSITION(646:646) CHAR(1),
  INIT_UTK_SECL POSITION(651:658) CHAR(8),
  INIT_UTK_EXECNODE POSITION(660:667) CHAR(8),
  INIT_UTK_SUSER_ID POSITION(669:676) CHAR(8),
  INIT_UTK_SNODE POSITION(678:685) CHAR(8),
  INIT_UTK_SGRP_ID POSITION(687:694) CHAR(8),
  INIT_UTK_SPOE POSITION(696:703) CHAR(8),
  INIT_UTK_SPCLASS POSITION(705:712) CHAR(8),
  INIT_UTK_USER_ID POSITION(714:721) CHAR(8),
  INIT_UTK_GRP_ID POSITION(723:730) CHAR(8),
  INIT_UTK_DFT_GRP POSITION(732:732) CHAR(1),
  INIT_UTK_DFT_SECL POSITION(737:737) CHAR(1),
  INIT_APPC_LINK POSITION(742:757) CHAR(16),
  INIT_UTK_NETW POSITION(759:766) CHAR(8),
  INIT_RES_NAME POSITION(768:1022) VARCHAR(255),
  INIT_CLASS POSITION(1024:1031) CHAR(8),
  INIT_X500_SUBJECT POSITION(1033:1287) VARCHAR(255),
  INIT_X500_ISSUER POSITION(1289:1543) VARCHAR(255),
  INIT_SERVSECL POSITION(1545:1552) CHAR(8),
  INIT_SERV_POENAME POSITION(1554:1617) CHAR(64),
  INIT_CTX_USER POSITION(1619:2128) CHAR(510),
  INIT_CTX_REG POSITION(2130:2384) CHAR(255),
  INIT_CTX_HOST POSITION(2386:2513) CHAR(128),
  INIT_CTX_MECH POSITION(2515:2530) CHAR(16)
)

```

Figure 13. DB2 Utility Statements Required to Load the Tables

Reorganizing the unloaded RACF SMF data in the DB2 database

Queries are processed faster if they are performed against an organized database. The DB2 utility statement required to reorganize the database is:

```
REORG TABLESPACE databasename.tablespace
```

Creating optimization statistics for the DB2 database

Queries are processed faster if they are performed against an organized database for which DB2 has collected performance statistics. The DB2 utility statement required to create these statistics is:

```
RUNSTATS TABLESPACE databasename.tablespace
```

Deleting data from the DB2 database

Before you reload the database with new data, you should archive the SMF flat files for future reference, and then delete the old data. Deleting the data can be done in several ways:

1. Use the DROP TABLE statement for each table you want to delete.
2. Use the DROP TABLESPACE statement for each table space.
3. Delete all of the records in each table.

Figure 14 shows the sample SQL statements that delete the group record data from the tables.

```
DELETE FROM userid.JOBINIT;  
DELETE FROM userid.ACCESS;  
DELETE FROM userid.ADDVOL;  
DELETE FROM userid.RENAMEDS;  
DELETE FROM userid.DELRES;
```

Figure 14. DB2 Utility Statements Required to Delete the Group Records

DB2 table names

Member IRRADULD in SYS1.SAMPLIB creates DB2 tables for each record type. Table 5 provides a useful reference of record type, record name, and DB2 table name.

Table 5. Correlation of DB2 Table Names and Record Types

DB2 Table Name	Column Prefix	Description
JOBINIT	INIT	Job initiation
ACCESS	ACC	Resource access, other than file or directory
ADDVOL	ADV	ADDVOL/CHGVOL
RENAMEDS	REN	Rename data set
DELRES	DELR	Delete resource
DELVOL	DELV	Delete volume
DEFINE	DEF	Define resource
ADDSD	AD	ADDSD command
ADDGROUP	AG	ADDGROUP command
ADDUSER	AU	ADDUSER command
ALTDSD	ALD	ALTDSD command
ALTGROUP	ALG	ALTGROUP command

Table 5. Correlation of DB2 Table Names and Record Types (continued)

DB2 Table Name	Column Prefix	Description
ALTUSER	ALU	ALTUSER command
CONNECT	CON	CONNECT command
DELDSD	DELD	DELDSD command
DELGROUP	DELG	DELGROUP command
DEUSER	DELU	DEUSER command
PASSWORD	PWD	PASSWORD command
PERMIT	PERM	PERMIT command
RALTER	RALT	RALTER command
RDEFINE	RDEF	RDEFINE command
RDELETE	RDEL	RDELETE command
REMOVE	REM	REMOVE command
SETROPTS	SETR	SETROPTS command
RVARY	RVAR	RVARY command
APPCLU	APPC	APPC session
GENERAL	GEN	General purpose
DIRSRCH	DSCH	Directory search
DACCESS	DACC	Check access to a directory
FACCESS	FACC	Check access to file
CHAUDIT	CAUD	Change audit options
CHDIR	CDIR	Change current directory
CHMOD	CMOD	Change file mode
CHOWN	COWN	Change file ownership
CLRSETID	CSID	Clear SETID bits for a file
EXESETID	ESID	EXEC with SETUID/SETGID
GETPSENT	GPST	Get z/OS UNIX process entry
INITOEDP	IOEP	Initialize z/OS UNIX process
TERMOEDP	TOEP	z/OS UNIX process complete
KILL	KILL	Terminate a process
LINK	LINK	LINK
MKDIR	MDIR	Make directory
MKNOD	MNOD	Make node
MNTFSYS	MFS	Mount a file system
OPENFILE	OPEN	Open a new file
PTRACE	PTRC	PTRACE authority checking
RENAMEF	RENF	Rename file
RMDIR	RDIR	Remove directory
SETEGID	SEGI	Set effective GID
SETEUID	SEUI	Set effective UID
SETGID	SGI	Set GID
SETUID	SUI	Set UID
SYMLINK	SYML	SYMLINK
UNLINK	UNL	UNLINK
UMNTFSYS	UFS	Unmount file system
CHKFOWN	CFOW	Check file owner
CHKPRIV	CPRV	Check z/OS UNIX privilege

Table 5. Correlation of DB2 Table Names and Record Types (continued)

DB2 Table Name	Column Prefix	Description
OPENSTTY	OSTY	Open slave TTY
RACLINK	RACL	RACLINK command
IPCCHK	ICLK	Check IPC access
IPCGET	IGET	Make ISP
IPCCTL	ICTL	R_IPC control
SETGROUP	SETG	Set group
CKOWN2	CKO2	Check owner two files
ACCR	ACCR	Access rights passed
RACDCERT	RACD	RACDCERT command
RACFINIT	RINI	RACF initialization data
CLASNAME	RINC	RACF class data
DSNSAFF	DSAF	Data sets affected by a security label change
INITACEE	INTA	initACEE functions
KTICKET	KTKT	Kerberos ticket
PDACCESS	PDAC	Policy Director Access Control Decision
PKIDPUBR	PKDP	CRL Publication
RPKIGENC	RPKG	R_PKIServ GENCERT
RPKIEXPT	RPKE	R_PKIServ EXPORT
RPKIREAD	RPKR	R_PKIServ Read Data
RPKISCEP	RPKS	R_PKIServ SCEP Request
RPKIUPDR	RPKU	R_PKIServ Update Request
RPKIUPDC	RPKC	R_PKIServ Update Certificate
SETFACL	SACL	R_Setfacl SETFACL
DELFACL	DACL	R_Setfacl DELFACL
SETFSECL	SSCL	R_setfsecl Set file security label
WRITEDWN	WDWN	R_writepriv Set writedown privilege
RPKIRESP	RPKO	R_PKIServ Respond
PTCREATE	PTCR	PassTicket Creation
PTEVAL	PTEV	PassTicket Evaluation
RDATAUPD	RPUT	R_Datalib
PKIAURNW	PKRN	PKI AutoRenew
PGMVERYF	PGMV	R_PgmSignVer
RACMAP	RACM	RACMAP command
AUTOPROF	AUTO	getUMAP/getGMAP automatic profile update
RPKIQREC	RPKQ	R_PKIServe QRECOVER

Using the RACF SMF data unload utility to generate XML documents

The records produced by the SMF data unload utility can be formatted as an Extensible Markup Language (XML) document. XML has many advantages over the usual tabular-style data, such as the many applications that can use XML as a format for reading and writing of data. The benefits of XML include:

- A better view of the data. Instead of the tabular format, which may be difficult to focus in on the information you are looking for, the XML audit report formats the data for ease of reading and retrieval.
- The display can include different fonts, text emphasis (bold, italic) and different colors to differentiate information.
- A complete set of data for each field. The tabular data is limited by space and can be truncated. XML does not have this restriction.
- A view of the audit data that can be tailored to your environment.

XML overview

XML is a flexible language which allows you to tag data and have it displayed in a variety of ways. Many software applications read and write XML data, both in enterprise computing and consumer applications. Therefore, an auditing report using XML can be distributed and analyzed on multiple platforms and operating systems. For more information about XML, see <http://www.ibm.com/servers/eserver/zseries/software/xml/>. For hints and tips on XML, see <http://www.ibm.com/developerworks/xml/library/x-tips.html>.

An XML document which contains the audit report looks like this:

```
<?xml version='1.0'?>
<securityEventLog xmlns='http://www.ibm.com/xmlns/zOS/IRRSchema'>

  <rdf:Description rdf:about=''
    xmlns:rdf='http://www.w3.org/1999/02/22-rdf-syntax-ns#'
    xmlns:dc='http://purl.org/dc/elements/1.1/'
    xmlns:z='http://www.ibm.com/xmlns/zOS'>

    <dc:creator>
      <z:application>SMF Unload</z:application>
      <z:product>z/OS Security Server RACF</z:product>
      <z:fmid>HRF7720</z:fmid>
    </dc:creator>
    <dc:subject>RACF Security Event Log 2003-01-01 04:12:33</dc:subject>
    <dc:language>en</dc:language>
  </rdf:Description>

  <event>
    <eventType>*CONNECT</eventType>
    <eventQual>SUCCESS</eventQual>
    <timeWritten>02:03:01.23</timeWritten>
    <dateWritten>2004-03-28</dateWritten>
    <systemSmfid>SYSA</systemSmfid>
    <prodName>Enterprise Identity Mapping</prodName>
    <prodFmid>HRF7720</prodFmid>
    <details xmlns:d='http://www.ibm.com/xmlns/zOS/EIMSchema'>
      <violation>Y</violation>
      <userNdfnd>Y</userNdfnd>
      <userWarning>Y</userWarning>
      <evtUserId>IBMUSER</evtUserId>
      <evtGrpId>SYS1</evtGrpId>
      <authNormal>Y</authNormal>
      <authSpecial>Y</authSpecial>
      <authOper>Y</authOper>
      <authAudit>Y</authAudit>
      <authExit>Y</authExit>
      <authFailsft>Y</authFailsft>
      <authBypass>Y</authBypass>
      <authTrusted>Y</authTrusted>
      <logClass>Y</logClass>
      <logUser>Y</logUser>
      <logSpecial>Y</logSpecial>
      <logAccess>Y</logAccess>
      <logRacinit>Y</logRacinit>
      <logAlways>Y</logAlways>
      <logCmdviol>Y</logCmdviol>
      <logGlobal>Y</logGlobal>
      <termLevel>934</termLevel>
      <backoutFail>Y</backoutFail>
      <profSame>Y</profSame>
      <term>L0437634</term>
      <jobName>$EIMTEST</jobName>
      <readTime>01:03:04</readTime>
      <readDate>2004-03-28</readDate>
      <smfUserId>SMFUSER</smfUserId>
      <logLevel>Y</logLevel>
      <logLogopt>Y</logLogopt>
      <logSec1>Y</logSec1>
      <logCompatm>Y</logCompatm>
    </details>
  </event>
</securityEventLog>
```

```

<logApp1aud>Y</logApp1aud>
<usrSec1>HIGHEST</usrSec1>
<logVmevent>Y</logVmevent>
<logNonomvs>Y</logNonomvs>
<logOmvsnprv>Y</logOmvsnprv>
<authOmvsu>Y</authOmvsu>
<authOmvsys>Y</authOmvsys>
<racfVersion>7720</racfVersion>
<srvrUserId>IBMUSER</srvrUserId>
<srvrGrpId>SYS1</srvrGrpId>
<prodId>EIM</prodId>
<logRauditx>Y</logRauditx>
<x500Subject>cn=ibmuser,c=us</x500Subject>
<x500Issuer>cn=PKI CA,c=us</x500Issuer>
<resName>EIM.MYDOMAIN.CONNECT</resName>
<class>RAUDITX</class>
<profileName>EIM.*.CONNECT</profileName>
<d:api>eimConnect</d:api>
<d:domainUrl>ldap://some.big.host/ibm-eimdomainname=My Domain, c=us</d:domainUrl>
<d:connectType>SIMPLE</d:connectType>
<d:bindUser>cn=EIM administrator</d:bindUser>
<d:certLabel>label</d:certLabel>
<d:keyRing>keyring</d:keyRing>
</details>
</event>

```

Producing XML output

You can have SMF Unload create an XML document by including only one of the following DD names in your IFASMFDP job:

- XMLOUT DD creates a compressed form of the XML document
- XMLFORM DD creates a more readable form of the XML document.

Whichever output type is coded in the JCL is the only type that is generated. For instance:

- If XMLFORM DD is specified in the JCL, then only the expanded, more readable XML output is written
- If XMLOUT DD is specified in the JCL, then only the compressed XML output type is written
- If OUTDD DD is specified in the JCL, then this is the only output type written (this is the traditional tabular non-xml form)
- If none of these types are specified in the JCL, the utility issues message IRR67522I **Open failed for OUTDD**

If more than one DDname is placed in the JCL, the above order (XMLFORM, XMLOUT, OUTDD) is used to see which one is created. The actual order of DD statements in the JCL is irrelevant.

For example:

```

//SMFDUMP EXEC PGM=IFASMFDP
//SYSPRINT DD SYSOUT=A
//ADUPRINT DD SYSOUT=A
//XMLOUT DD DISP=SHR,DSN=USER01.RACF.SMF.XMLOUT
//SMFDATA DD DISP=SHR,DSN=USER01.RACF.SMFDATA
//SMFOUT DD DUMMY
//SYSIN DD *
INDD(SMFDATA,OPTIONS(DUMP))
OUTDD(SMFOUT,TYPE(30(1,5),80:83))
ABEND(NORETRY)
USER2(IRRADU00)
USER3(IRRADU86)
//

```

This job creates a compressed form of the XML document in a data set that already exists. You can think of this as “raw output”, since it is the most basic form of the XML document. While this report takes up the least space, it is not well-suited for reading due to its limited line wrapping and tag justification. In the document, the

tags and information are often composed of one long line to save space. A more readable form of the report can be created using the XMLFORM DD statement. The output from this would include better line wrapping, and the tags would be justified so that they begin on new lines when necessary. It is a more readable form but takes up more space.

How the XML tag names are derived

The names of the tags and the syntax of the tags are defined by XML schema document. The schema can be used to validate the data contained in an XML document. The tags appear in the order described by the schema documents. The schema document for RACF can be found in Sys1.Samplib(IRRSCHEM).

In general, the tag names used in RACF are derived from the corresponding DB2 field names. The rules for converting a field name to a tag name are:

1. Remove the column name and the first underscore (“_”) from the field name
2. Capitalize the first letter after each of the remaining underscores in the name. The rest of the characters should be lowercase.
3. Remove the underscores from the name

The exceptions to this methodology are as follows:

Table 6. XML naming exceptions

DB2 Field Name	XML Tag Name
RINI_TERM	riniTerm
SECL_LINK	eventLink
CAUD_REQUEST_WRITE	caudRequestWrite
CAUD_REQUEST_READ	caudRequestRead
CAUD_REQUEST_EXEC	caudRequestExec
SSCL_OLDSECL	oldSecl
<col>logstring	logstr
KTKT_PRINCIPAL	kerbPrincipal
PDAC_PRINCIPAL	pdasPrincipal
any field with RESERVED in the name	Note: no XML tag
ACC_NAME	profileName
APPC_NAME	profileName

XML interprets certain characters as having a special meaning, such as "<" and ">". If a value contains one of these special characters, which are listed in Table 8 on page 69, SMF Unload replaces the value with an “entity reference” so that it will not be misinterpreted by an XML parser. Here is an example:

Table 7. XML interpretation of special characters example

Before Value	After Value
<subjectDN>cn=John,ou=Smith & Sons,c=us</subjectDN>	<subjectDN>cn=John,ou=Smith & Sons,c=us,<subjectDN>

The special characters are:

Table 8. XML special characters substitutions

Character	Substitution symbol
<	<
&	&
>	>
“	&odq;
'	'

It is possible for a single element or value in the XMLOUT or XMLFORM to cause the length of a record to exceed the maximum 8K limit. SMF Unload breaks the line into two. If the line break would naturally occur in the middle of a tag or entity reference, SMF Unload splits the line before or after the tag or entity reference so that the tag or entity reference is not broken. What this means is that the data value may include a carriage return or line feed that was not originally part of the value. It is up to the application processing the document to detect this condition and concatenate the two lines before passing the element to an XML parser.

Viewing and working with XML audit reports

On z/OS, you can process the document using the IBM XML Toolkit for z/OS. The XML can be used in the following ways:

- Viewed using the ISPF edit function
- Viewed using an XML-capable web browser
- Converted to HTML using a style sheet
- Processed by an XML parser and processor

On other systems, such as personal computers and workstations, the audit report can be viewed using an XML-capable web browser. Many browsers available today have the ability to correctly parse and render XML documents. Therefore, once the audit report is on that system, you can read it as easily as any other web document. Simply bring up a listing of the files and single- or double-click the file to open it in the browser window. The platform documentation can help you discover which applications are able to parse and display XML files.

One thing to note is that to use the XML file on a personal computer, you must first alter the EBCDIC encoding line at the top of the file:

```
<?xml version='1.0' encoding='ebcdic-cp-us' ?>
```

So that it looks like the following:

```
<?xml version='1.0' encoding='ISO8859-1' ?>
```

Event code qualifiers

The RACF event code (found in the SMF80EVT field of the SMF record) and the RACF event code qualifier (found in the SMF80EVQ field of the SMF record) are determined during RACF processing. The following sections explain the meaning of each qualifier code by event.

You may also see event codes for another component, such as EIM. The data in the audit record identifies the component. See the documentation of the component for details.

Event 1(1): JOB INITIATION/TSO LOGON/TSO LOGOFF

This event is logged by RACROUTE REQUEST=VERIFY and RACROUTE REQUEST=VERIFYX. Installation exit ICHRIX02 can change the return code of the RACROUTE REQUEST=VERIFY or RACROUTE REQUEST=VERIFYX request to any value. The return code significantly influences the corresponding audit record's event code 1 qualifier. You should be familiar with any ICHRIX02 processing in effect for your installation. See *z/OS Security Server RACF System Programmer's Guide* for details.

For this event, code qualifiers 0 and 8 do not exist as type 80 records. They are contained in the unloaded records from the RACF SMF data unload utility (IRRADU00) and as reports and reformatted records from the RACF report writer (RACFRW).

The explanations of the event code qualifiers for Event 1 are:

- 0(0) SUCCESSFUL INITIATION** The job began successfully.
- 1(1) INVALID PASSWORD** The password specified on the job card or at logon is incorrect.
- 2(2) INVALID GROUP** The user tried to log on or to initiate a job using a group that the user is not a member of.
- 3(3) INVALID OIDCARD** Operator identification cards are used at the installation, and the data received from the one used does not match that of the user's profile.
- 4(4) INVALID TERMINAL/CONSOLE** The user is not authorized to the port of entry (POE). There are four kinds of POEs, each with its own profile class: APPCPOR, CONSOLE, JESINPUT, and TERMINAL. One of the following occurred:
 - The port of entry is active but the user is not authorized.
 - The user is denied access because of conditional days/times in the user profile.
 - The user is denied access because of conditional days/times in the class profile (TERMINAL class only).
- 5(5) INVALID APPLICATION** The APPL class is active, and the user is trying to log on to an application without authorization.
- 6(6) REVOKED USER ID ATTEMPTING ACCESS** The user ID specified on the logon or job card has been revoked. One of the following occurred:
 - The installation-defined limit of password attempts was reached at an earlier time.
 - The inactive interval was reached.
 - The revoke date in the user's profile is in effect.
 - The RACF administrator revoked the user ID.

The RACF administrator must reset the user ID before the user can log on again.
- 7(7) USER ID AUTOMATICALLY REVOKED** The user ID has been automatically revoked. The installation-defined limit of password and password phrase attempts was reached.
- 8(8) SUCCESSFUL TERMINATION** The job completed successfully.

- 9(9) UNDEFINED USER ID** The user ID specified on the job card or at logon is not defined to the RACF database.
- 10(A) INSUFFICIENT SECURITY LABEL AUTHORITY** One of the following occurred:
- SETROPTS MLS FAILURES is in effect and the user's security label does not dominate the submitter's security label. Two exceptions are explained under Qualifier 20.
 - SETROPTS MLACTIVE FAILURES is in effect and the job card/logon attempt does not specify a valid security label. One exception is explained under Qualifier 21.
- 11(B) NOT AUTHORIZED TO SECURITY LABEL** The user is not authorized to the security label specified. One exception is explained under Qualifier 22.
- 12(C) SUCCESSFUL RACINIT INITIATION** The job or user was verified.
- 13(D) SUCCESSFUL RACINIT DELETE** The job completed or the user logged off.
- 14(E) SYSTEM NOW REQUIRES MORE AUTHORITY** SETROPTS MLQUIET is in effect. If this is a user verification, the user is not a console operator and does not have the SPECIAL attribute. If this is a job verification, the job is not part of the trusted computing base (TCB). The verification fails.
- 15(F) REMOTE JOB ENTRY—JOB NOT AUTHORIZED** The submitting node is not authorized to the system; a NODES profile prevents remote job entry. The profile has the format 'submit_node.RUSER.userid' and has a UACC of NONE.

Note:

Surrogate Function Qualifiers:

Qualifiers 16, 17, and 18 involve the use of the surrogate function, and occur if any of the following conditions is met:

- The SURROGAT class is active.
- General resource profiles of the SURROGAT class are defined for the job card's user ID, and the user ID submitting the job is permitted to the profile with at least READ access.
- The submitter is authorized to the security label of the job.

For more information, see *z/OS Security Server RACF Security Administrator's Guide*.

- 16(10) SURROGATE CLASS IS INACTIVE** The SURROGAT class is inactive. The job card has a user ID that is different from the submitter's user ID, and there is no password specified.
- 17(11) SUBMITTER IS NOT AUTHORIZED BY USER** The SURROGAT class is active. Either there is no SURROGAT profile for the job card's user ID, or the submitter's user ID is not permitted to the profile.
- 18(12) SUBMITTER IS NOT AUTHORIZED TO SECURITY LABEL** The SECLABEL class is active and there is a security label on the job card. The submitter is not authorized to the security label specified on the job card.
- 19(13) USER IS NOT AUTHORIZED TO JOB** The JESJOBS class is active, and the user is not authorized to the jobname.

- 20(14) WARNING—INSUFFICIENT SECURITY LABEL AUTHORITY** One of the following occurred:
- SETROPTS MLS WARNING is in effect and the security label on the job card does not dominate the submitter's security label.
 - SETROPTS MLS FAILURES is in effect, the user's security label does not dominate the submitter's, and the user has the SPECIAL attribute.
 - SETROPTS MLS FAILURES and SETROPTS COMPATMODE are in effect, the user's security label does not dominate the submitter's, and the submitter's or the job owner's security label is the default.

The verification does not fail.

- 21(15) WARNING—SECURITY LABEL MISSING FROM JOB, USER, OR PROFILE** One of the following occurred:

- MLACTIVE WARNING is in effect, and the job card or logon attempt did not specify a valid security label.
- MLACTIVE FAILURES is in effect, the user has the SPECIAL attribute, and a valid security label is not specified.

The verification does not fail.

- 22(16) WARNING—NOT AUTHORIZED TO SECURITY LABEL** The user has the SPECIAL attribute, the security label is SYSHIGH, and the user does not have authority to it. The verification does not fail.

- 23(17) SECURITY LABELS NOT COMPATIBLE** SETROPTS MLS is not active, the submitter's user ID is different from the user ID on the job card, and the submitter's and the user's security labels are disjoint (neither one dominates the other).

One exception is listed under Qualifier 24.

- 24(18) WARNING—SECURITY LABELS NOT COMPATIBLE** SETROPTS MLS is not active, the submitter's user ID is different from the user ID on the job card, the submitter's and user's security labels are disjoint, SETROPTS COMPATMODE is in effect, and the submitter's or user's security label is the default. The verification does not fail.

- 25(19) CURRENT PASSWORD HAS EXPIRED** The user's password has expired for one of the following reasons:

- The installation specification in SETROPTS PASSWORD INTERVAL command
- Creation of the password in the ADDUSER command
- Alteration of the password with the ALTUSER PASSWORD command

- 26(1A) INVALID NEW PASSWORD** The new password specified may be incorrect because:

- It is all blanks.
- The characters are not all alphanumeric.
- The characters do not match the installation's password syntax rules (set by the SETROPTS PASSWORD command).
- It is the same as a past password (the extent of the past history determined by the SETROPTS PASSWORD HISTORY command).
- It is marked invalid by the installation's password exit.
- It is too soon to change the password (as determined by the SETROPTS PASSWORD MINCHANGE command).

- 27(1B) VERIFICATION FAILED BY INSTALLATION** The installation exit ICHRIX01 or ICHRIX02 failed the request.
- 28(1C) GROUP ACCESS HAS BEEN REVOKED** The user's membership to the group specified has been revoked.
- 29(1D) OIDCARD IS REQUIRED** An OI DCARD is required by the installation but none was given.
- 30(1E) NETWORK JOB ENTRY—JOB NOT AUTHORIZED** For session types of NJE SYSOUT or NJE BATCH, the verification fails because one of the following occurred:
- The user, group, or security label requirements in the NODES profiles were not met.
 - The submitter's node is not valid.
 - The reverify check failed.
- See *z/OS Security Server RACF Security Administrator's Guide* for details on NJE.
- 31(1F) WARNING—UNKNOWN USER FROM TRUSTED NODE PROPAGATED** The combination of having a trusted node submit a job with the undefined user ID warrants this logging. The verification does not fail.
- For an NJE BATCH job, the submitting user is the NJE undefined user ID. The default NJE undefined user ID is eight question marks (????????), unless it was changed with the SETROPTS JES NJEUSERID command. The submitting node is trusted (its best-fit NODES profile on the receiving node's system has a UACC of at least UPDATE). This profile allows propagation of submitters; however, the undefined user ID does not propagate.
- 32(20) SUCCESSFUL INITIATION USING PASSTICKET** Logon was achieved using a PassTicket.
- 33(21) ATTEMPTED REPLAY OF PASSTICKET** Logon was rejected because of attempted replay of a PassTicket.
- 34(22) CLIENT SECURITY LABEL NOT EQUIVALENT TO SERVER'S** Logon was rejected because security labels are not equivalent.
- 35(23) USER AUTOMATICALLY REVOKED DUE TO INACTIVITY** A user has not logged on, submitted a job or accessed the system for so long that the user ID has become inactive. RACF prevents the user from accessing the system.
- 36(24) PASS 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. RACF prevents the user from accessing the system.
- 37(25) NEW PASS PHRASE IS NOT VALID** Logon was rejected because the new password phrase is not valid.
- 38(26) CURRENT PASS PHRASE HAS EXPIRED** Logon was rejected because the current password phrase has expired.
- 39(27) NO RACF USER ID FOUND FOR DISTRIBUTED IDENTITY** Logon was rejected because no RACF user ID was found for the distributed identity.

Event 2(2): RESOURCE ACCESS

This event is logged by RACROUTE REQUEST=AUTH, RACROUTE REQUEST=DIRAUTH and RACROUTE REQUEST=FASTAUTH.

The explanations of the event code qualifiers for Event 2 are:

- 0(0) **SUCCESSFUL ACCESS** The user has authorization to the resource.
- 1(1) **INSUFFICIENT AUTHORITY** The user does not have authorization to the resource.
- 2(2) **PROFILE NOT FOUND—RACFIND SPECIFIED ON MACRO** If the request is AUTH, the RACFIND keyword equaled YES on the authorization request, specifying that a discrete profile should exist for the resource. No discrete or generic RACF protection was found.

If the request is FASTAUTH, the program is not controlled and the PADS data sets are open.
- 3(3) **ACCESS PERMITTED DUE TO WARNING** The user does not have proper authority to the resource. However, the resource's profile has the WARNING option and allows the access.

Note:

Exceptions

- PROGRAM class profiles cannot use the WARNING option.
- RACLISTed profiles use the WARNING option only if they are RACLISTed by SETROPTS or a RACROUTE REQUEST=LIST that specifies RELEASE=1.8 or later.

- 4(4) **FAILED DUE TO PROTECTALL SETROPTS PROTECTALL FAILURES** is in effect, and the data set has not been protected by a discrete or generic profile.

Note:

Exceptions

- A privileged user bypasses this checking (no auditing done).
- A trusted user bypasses the checking, but can be audited with the SETROPTS LOGOPTIONS command.
- A user with the SPECIAL attribute gets a warning (see Qualifier 5).
- A system-generated temporary data set does not require protection.

- 5(5) **WARNING ISSUED DUE TO PROTECTALL SETROPTS PROTECTALL WARNING** is in effect, and the data set has not been protected by a discrete or generic profile. The authorization request does not fail.

The exceptions in Qualifier 4 also apply.

- 6(6) **INSUFFICIENT CATEGORY/SECLEVEL** The installation uses categories or security levels as separate entities. One of the following occurred:
 - The user's SECLEVEL is less than the SECLEVEL of the resource.
 - The user is not a member of every CATEGORY associated with the resource.
- 7(7) **INSUFFICIENT SECURITY LABEL AUTHORITY** The SECLABEL class is active and one of the following occurred:

- The user's security label does not dominate the resource's.
- The user does not have a security label, but the resource does.
- SETROPTS MLACTIVE FAILURES is in effect, and either the user or the resource is missing a security label. One exception is explained in Qualifier 8.
- The resource's class requires reverse domination checking, and the resource's security label does not dominate the user's.
- SETROPTS MLS FAILURES is in effect; the user's security label does not equal the resource's, and the requested access is UPDATE or CONTROL. One exception is explained under Qualifier 9.

8(8) SECURITY LABEL MISSING FROM JOB, USER OR PROFILE One of the following occurred:

- SETROPTS MLACTIVE WARNING is in effect, the SECLABEL class is active, and either the resource or user is missing a security label.
- SETROPTS MLACTIVE FAILURES is in effect, the user has the SPECIAL attribute, and either the resource or the user is missing a security label.

9(9) WARNING—INSUFFICIENT SECURITY LABEL AUTHORITY One of the following occurred:

- The SECLABEL class is active, SETROPTS MLS WARNING is in effect, the user's security label does not equal the resource's security label, and the requested access is UPDATE or CONTROL.
- SETROPTS MLS FAILURES is in effect, the user's security label does not equal the resource's security label, the requested access is UPDATE or CONTROL, and the user has the SPECIAL attribute.

10(A) WARNING—DATA SET NOT CATALOGED SETROPTS CATDSNS WARNING is in effect. The data set being accessed cannot be cataloged.

See *z/OS Security Server RACF Command Language Reference* for more information.

11(B) DATA SET NOT CATALOGED SETROPTS CATDSNS FAILURES is in effect. The data set being accessed cannot be cataloged. If the user has the SPECIAL attribute, only a warning is issued (see Qualifier 10).

See *z/OS Security Server RACF Command Language Reference* for more information.

12(C) PROFILE NOT FOUND—REQUIRED FOR AUTHORITY CHECKING A profile was not found for the general resource, and that resource's class has a default return code greater than 4. The authorization request fails.

13(D) WARNING—INSUFFICIENT CATEGORY/SECLEVEL The installation uses categories or security levels as separate entities. One of the following occurred:

- The user's SECLEVEL is less than the SECLEVEL of the resource.
- The user is not a member of every CATEGORY associated with the resource.

The resource profile has the WARNING option, so access is given.

Note:

Exceptions

- PROGRAM class profiles cannot use the WARNING option.

- RACLISTed profiles can use the WARNING option only if they are RACLISTed by SETROPTS or a RACF 1.8 (or later) RACROUTE REQUEST=LIST.
- 14(E) WARNING—NON-MAIN EXECUTION ENVIRONMENT** Non-MAIN execution environment was detected while in ENHANCED PGMSECURITY mode. Conditional access for Program Access to Data Sets (PADS) or access to EXECUTE-controlled program is temporarily allowed.
- 15(F) CONDITIONAL ACCESS ALLOWED VIA BASIC MODE PROGRAM** Conditional access for Program Access to Data Sets (PADS) or access to EXECUTE-controlled program is allowed through the BASIC mode program while in ENHANCED PGMSECURITY mode.

Event 3(3): ADDVOL/CHGVOL

This event refers to RACROUTE REQUEST=DEFINE,TYPE=ADDVOL and RACROUTE REQUEST=DEFINE,TYPE=CHGVOL.

The explanations of the event code qualifiers for Event 3 are:

- 0(0) SUCCESSFUL PROCESSING OF NEW VOLUME** One of the following occurred:
- The user has proper administrative authority to the DATASET profile; in the case of tape data sets with TAPEVOL active, the user also had administrative authority to the TAPEVOL profile.
 - SETROPTS MLS WARNING is in effect, the TAPEVOL class is active, a TAPEVOL profile exists, and the user's security label does not equal the resource's.
 - SETROPTS MLACTIVE WARNING is in effect, the TAPEVOL class is active, and no TAPEVOL profile exists for the volume.
- 1(1) INSUFFICIENT AUTHORITY** The user did not have administrative authority to the DATASET profile, or, in the case of tape data sets, the TAPEVOL class is active and the user did not have administrative authority to the TAPEVOL profile.
- 2(2) INSUFFICIENT SECURITY LABEL AUTHORITY** The SECLABEL class is active, the data set is a tape data set, the TAPEVOL class is active, and the user's security label does not dominate the security label found in the TAPEVOL profile.
- 3(3) LESS SPECIFIC PROFILE EXISTS WITH DIFFERENT SECURITY LABEL** The SECLABEL class is active, SETROPTS MLSTABLE is in effect, a less specific generic profile exists that does not have the same security label, the data set is a tape data set, and the TAPEVOL class is active. Changing the volume would change the TAPEVOL profile's security label, violating SETROPTS MLSTABLE rules.

Note:

Exception

If SETROPTS MLQUIET is also in effect and the user has the SPECIAL attribute, the request does not fail and this event is not logged.

Event 4(4): RENAME RESOURCE

This event is based on RACROUTE
REQUEST=DEFINE,TYPE=DEFINE,NEWNAME or RACROUTE
REQUEST=DEFINE,TYPE=DEFINE,NEWNAMX.

The explanations of the event code qualifiers for Event 4 are:

- 0(0) SUCCESSFUL RENAME** One of the following occurred:
- The user has sufficient authority to rename the resource.
 - The SECLABEL class is active, SETROPTS MLACTIVE WARNING is in effect, and the user or the resource does not have a security label.
- 1(1) INVALID GROUP** The resource to be renamed is a data set, and the high-level qualifier of the new data set is not a valid group or user ID.
- 2(2) USER NOT IN GROUP** The resource is a data set, RACFIND is not set to NO, the high-level qualifier of the new data set name is a group, and the user does not belong to that group.
- 3(3) INSUFFICIENT AUTHORITY** One of the following occurred:
- SETROPTS GENERICOWNER is in effect, and renaming the profile would violate GENERICOWNER rules.
 - The resource is a data set, and the high-level qualifier is a group or user ID. The user is not authorized to create a new data set by the generic profile protecting the new name, and the high-level qualifier of the new data set name is beyond the scope of the user.
 - The resource is an SFS file or directory, and the second qualifier is a user ID. The user is not authorized to create a new file or directory by the generic profile protecting the new name, and the second qualifier of the new file or directory name is beyond the scope of the user.
- See *z/OS Security Server RACF Security Administrator's Guide*.
- 4(4) RESOURCE NAME ALREADY DEFINED** The requested new name already has a discrete profile defined. The return code of the RENAME is 4.
- 5(5) USER NOT DEFINED TO RACF** The installation's naming convention routine has indicated that the high-level qualifier is a user ID that is not defined to RACF. One of the following occurred:
- RACFIND is not set to NO.
 - The resource is protected by a generic or global profile, and the user does not have ALTER access to it.
- 6(6) RESOURCE NOT PROTECTED** SETROPTS PROTECTALL FAILURES is in effect, and the new data set name is not protected by a profile.
- 7(7) WARNING—RESOURCE NOT PROTECTED** SETROPTS PROTECTALL WARNINGS is in effect, and the new data set name is not protected by a profile.
- The RENAME is allowed.
- 8(8) USER IN SECOND QUALIFIER IS NOT RACF DEFINED** The second qualifier of the new name is not a valid user ID.
- 9(9) LESS SPECIFIC PROFILE EXISTS WITH DIFFERENT SECURITY LABEL** The SECLABEL class is active, SETROPTS MLSTABLE is in effect,

and there is a less specific generic profile existing for the new name with a different security label. Renaming this resource would violate SETROPTS MLSTABLE rules.

- 10(A) **INSUFFICIENT SECURITY LABEL AUTHORITY** The SECLABEL class is active, SETROPTS MLS FAILURES is in effect, and the user is not authorized to the security label of the resource to be renamed.
- 11(B) **RESOURCE NOT PROTECTED BY SECURITY LABEL** The SECLABEL class is active, SETROPTS MLS FAILURES is in effect, and the profile covering the old resource name does not have a security label.
- 12(C) **NEW NAME NOT PROTECTED BY SECURITY LABEL** The SECLABEL class is active, SETROPTS MLS FAILURES is in effect, and the profile that would cover the new resource name does not have a security label.
- 13(D) **NEW SECURITY LABEL MUST DOMINATE OLD SECURITY LABEL** The SECLABEL class is active, SETROPTS MLS FAILURES is in effect, and the security label of the profile covering the new resource name does not dominate the security label of the profile covering the old resource name.
- 14(E) **INSUFFICIENT SECURITY LABEL AUTHORITY** The SECLABEL class is active, SETROPTS MLS WARNING is in effect, and the user is not authorized to the security label of the profile. The RENAME is allowed.
- 15(F) **WARNING—RESOURCE NOT PROTECTED BY SECURITY LABEL** The SECLABEL class is active, SETROPTS MLS WARNING is in effect, and the profile covering the old resource name does not have a security label. The RENAME is allowed.
- 16(10) **WARNING—NEW NAME NOT PROTECTED BY SECURITY LABEL** The SECLABEL class is active, SETROPTS MLS WARNING is in effect, and the profile that would cover the new resource name does not have a security label. The RENAME is allowed.
- 17(11) **WARNING—NEW SECURITY LABEL MUST DOMINATE OLD SECURITY LABEL** The SECLABEL class is active, SETROPTS MLS WARNING is in effect, and the security label of the profile covering the new resource name does not dominate the security label of the profile covering the old resource name. The RENAME does not fail.

Event 5(5): DELETE RESOURCE

This event is based on RACROUTE REQUEST=DEFINE,TYPE=DELETE.

The explanations of the event code qualifiers for Event 5 are:

- 0(0) **SUCCESSFUL SCRATCH** The resource profile was deleted.
- 1(1) **RESOURCE NOT FOUND** The resource profile was not found.
- 2(2) **INVALID VOLUME** The class is DATASET, and the data set does not reside on the volume specified.

Event 6(6): DELETE ONE VOLUME OF A MULTIVOLUME RESOURCE

This event is based on RACROUTE REQUEST=DEFINE,TYPE=DELETE.

The explanations of the event code qualifiers for Event 6 are:

- 0(0) **SUCCESSFUL DELETION** The volume was successfully deleted from the DATASET profile.

Event 7(7): DEFINE RESOURCE

This event is based on RACROUTE REQUEST=DEFINE,TYPE=DEFINE.

The explanations of the event code qualifiers for Event 7 are:

0(0) SUCCESSFUL DEFINITION

- The user had sufficient authority to define the resource.
- The SECLABEL class is active, SETROPTS MLACTIVE WARNING is in effect, and the user or the resource does not have a security label.

1(1) GROUP UNDEFINED The resource to be defined is a data set, and the high-level qualifier is not a valid group or user ID.

2(2) USER NOT IN GROUP The resource is a data set, RACFIND is not set to NO, the high-level qualifier is a group, and the user does not belong to that group.

3(3) INSUFFICIENT AUTHORITY One of the following occurred:

- SETROPTS GENERICOWNER is in effect and defining the profile would violate GENERICOWNER rules.
- For general resources, the user is not authorized to define profiles in the class.
- The resource is a data set, and the high-level qualifier of the resource is a group or user ID. The user is not authorized to create a new data set by the generic profile protecting the new name, and the high-level qualifier of the new data set name is beyond the scope of the user.
- The resource is an SFS file or directory, and the second qualifier is a user ID. The user is not authorized to create a new file or directory by the generic profile protecting the new name, and the second qualifier of the new file or directory name is beyond the scope of the user.

See *z/OS Security Server RACF Security Administrator's Guide*.

4(4) RESOURCE NAME ALREADY DEFINED The requested name already has a discrete profile defined. The return code of the DEFINE is 4.

5(5) USER NOT DEFINED TO RACF The installation's naming convention routine has indicated that the high-level qualifier is a user ID that is not defined to RACF. One of the following occurred:

- RACFIND is not set to NO.
- The resource is protected by a generic or global profile, and the user does not have ALTER access to it.

6(6) RESOURCE NOT PROTECTED SETROPTS PROTECTALL FAILURES is in effect, and the data set to be defined will not be protected by a profile.

7(7) WARNING—RESOURCE NOT PROTECTED SETROPTS PROTECTALL WARNINGS is in effect, and the data set to be defined will not be protected by a profile. The DEFINE is allowed.

8(8) WARNING—SECURITY LABEL MISSING FROM JOB, USER, OR PROFILE The SECLABEL and TAPEVOL classes are active. SETROPTS MLACTIVE WARNING is in effect, and the TAPEVOL profile is without a security label. The DEFINE is allowed.

9(9) INSUFFICIENT SECURITY LABEL AUTHORITY The SECLABEL and TAPEVOL classes are active. SETROPTS MLS WARNING is in effect, and the user's security label does not dominate the one found in the TAPEVOL profile.

The DEFINE is allowed.

- 10(A) USER IN SECOND QUALIFIER IS NOT RACF-DEFINED** The second qualifier of the name is not a valid user ID.
- 11(B) INSUFFICIENT SECURITY LABEL AUTHORITY** The SECLABEL class is active, and one of the following occurred:
- SETROPTS MLACTIVE FAILURES is in effect, and the user is missing a security label.
 - SETROPTS MLACTIVE FAILURES is in effect, and the resource is missing a security label.
 - The user's security label does not dominate the resource's.
 - SETROPTS MLS FAILURES is in effect, and the user's security label does not equal the resource's.
- 12(C) LESS SPECIFIC PROFILE EXISTS WITH A DIFFERENT SECURITY LABEL** The SECLABEL class is active, SETROPTS MLSTABLE is in effect, and there is a less specific generic profile existing for the name with a different security label.

Defining this resource would violate SETROPTS MLSTABLE rules.

Event 8(8)–25(19): COMMANDS

Events 8 through 25 apply to the RACF commands. The following qualifier codes are used for each event:

- 0(0) NO VIOLATIONS DETECTED** The RACF command was issued successfully. This qualifier applies to all RACF commands.
- 1(1) INSUFFICIENT AUTHORITY** The user did not have the authority to issue the RACF command. This qualifier applies to all RACF commands.
- 2(2) KEYWORD VIOLATIONS DETECTED** The user had the authority to issue the RACF command, but not to all the keywords that were specified. Keywords that the user is not authorized to use are ignored. For example, a user with the SPECIAL attribute but without the AUDITOR attribute can issue the ALTUSER command, but not with the GLOBALAUDIT keyword. This qualifier applies to all RACF commands.
- 3(3) SUCCESSFUL LISTING OF DATASETS** This logs the successful use of LISTDSD DSNS.
- 4(4) SYSTEM ERROR IN LISTING OF DATA SETS** This logs an error in attempting LISTDSD DSNS.

Note:

1. When the SETROPTS command is issued with a keyword that contains an asterisk (*), the asterisk is displayed in the output. For example, if you issue the command SETROPTS AUDIT(*), the output contains AUDIT(*).
2. When the SETROPTS command is issued with a keyword that lists more than ten classes, the output lists the first ten classes and displays the remaining number as an ellipsis. For example, if you issue the command SETROPTS CLASSACT(class1 class2 class3 class4 class5 class6 class7 class8 class9 class10 class11 class12), the output appears as CLASSACT(class1 class2 class3 class4 class5 class6 class7 class8 class9 class10 ...(00002)).
3. When the RVARY command is issued, the DATASET keyword lists the names of as many RACF databases as can fit in the 1024 character output. The remainder are shown as an ellipsis (...(nnnnn)).

4. When the RVAR Y command is issued with the NOCLASSACT(*) keyword or with more than ten classes specified, the output lists the first ten classes. The remaining classes are shown as an ellipsis.

Event 26(1A): APPCLU

This event is logged by RACROUTE REQUEST=AUDIT,EVENT='APPCLU'. This event applies to establishing a session between two logical units (referred to as the local LU and the partner LU) in accordance with the System Network Architecture (SNA). VTAM and CICS® call RACF for security information stored in general resource profiles in the APPCLU class.

Each profile contains an 8-byte session key that is used in verification; the two LUs must have corresponding profiles with identical keys so that the handshaking of encrypted data is successful.

The explanations of the event code qualifiers for Event 26 are:

- 0(0) **PARTNER VERIFICATION WAS SUCCESSFUL** The handshaking was successful. The LUs established a connection.
- 1(1) **SESSION ESTABLISHED WITHOUT VERIFICATION** No handshaking was done, but the LUs were still allowed to establish a connection, with the knowledge that the partners were not verified.
- 2(2) **LOCAL LU KEY WILL EXPIRE IN 5 DAYS OR LESS** The handshaking was successful; this qualifier was set to tell users when the local LU's session key would expire.
- 3(3) **PARTNER LU ACCESS HAS BEEN REVOKED** Too many unsuccessful attempts were made at matching the session key.
- 4(4) **PARTNER LU KEY DOES NOT MATCH THIS LU KEY** An attempt was made to establish a session, but the session keys did not match. For example, the two sets of identical data encrypted with the two keys did not match.
- 5(5) **SESSION TERMINATED FOR SECURITY REASONS** One or both of the APPCLU profiles involved have the keyword LOCK specified in their session information, preventing any connections from being made. This keyword enables the security administrator to temporarily prevent specific connections without deleting any profiles.
- 6(6) **REQUIRED SESSION KEY NOT DEFINED** The local LU had VERIFY=REQUIRED coded on its APPL statement, indicating that session level verification must be used on all sessions with the LU. One of the following occurred:
 - The local LU is the primary LU and no password was defined in RACF for the LU pair.
 - The partner LU is the primary LU, but the bind it sent to the local LU did not contain random data (which would indicate that the partner is using session level verification also).
- 7(7) **POSSIBLE SECURITY ATTACK BY PARTNER LU** The local LU sent out a random number to another LU as part of the handshaking process of establishing a session. That same number then came in from a third LU for the local LU to encrypt. It is a coincidence that the same number is chosen; the number is 64 bits of random data.

It may be that an unauthorized user is attempting to steal the encrypted response.

- 8(8) **SESSION KEY NOT DEFINED FOR PARTNER LU** The local LU had VERIFY=OPTIONAL coded on its APPL statement. There was a password defined in the local LU's RACF profile for the LU-LU pair, indicating that session level verification should be used on all sessions between the two LUs. However, the partner LU tried to start a session without using session level verification.
- 9(9) **SESSION KEY NOT DEFINED FOR THIS LU** The local LU had VERIFY=OPTIONAL coded on its APPL statement. No password was defined in the local LU's RACF profile for the LU-LU pair, indicating that session level verification may not be used to establish sessions with this LU. However, the partner LU tried to establish a session using session level verification.
- 10(A) **SNA SECURITY-RELATED PROTOCOL ERROR** The LU trying to establish a connection is not responding correctly according to the handshaking protocol.
- 11(B) **PROFILE CHANGE DURING VERIFICATION** The handshaking was attempted, but it is evident that one of the LU's profiles (specifically the session key) changed in the middle of the handshaking, making its success impossible.
- 12(C) **EXPIRED SESSION KEY** The session key in one or both of the APPCLU profiles has expired.

Event 27(1B): GENERAL AUDITING

This event is logged by RACROUTE REQUEST=AUDIT,EVENT='GENERAL'. RACF does not make any authority checks for this event.

The explanations of the event code qualifiers for Event 27 are:

0 - 99 GENERAL AUDIT RECORD WRITTEN

Qualifiers 0 to 99 can be used for Event 27. These qualifiers are installation defined.

Event 28(1C)–58(3A): z/OS UNIX EVENT TYPES

Events 28 through 58 apply to z/OS UNIX. The following qualifier codes are used for each event:

28(1C) DIRECTORY SEARCH

- 0(0) Access allowed
- 1(1) Not authorized to search directory
- 2(2) Security label failure

29(1D) CHECK ACCESS TO DIRECTORY

- 0(0) Access allowed
- 1(1) Caller does not have requested access authority
- 2(2) Security label failure

30(1E) CHECK ACCESS TO FILE

- 0(0) Access allowed

- 1(1) Caller does not have requested access authority
- 2(2) Security label failure
- 31(1F) CHAUDIT**
 - 0(0) File's audit options changed
 - 1(1) Caller does not have authority to change user audit options of specified file
 - 2(2) Caller does not have authority to change auditor audit options
 - 3(3) Security label failure
- 32(20) CHDIR**
 - 0(0) Current working directory changed
 - * Failures logged as directory search event types
- 33(21) CHMOD**
 - 0(0) File's mode changed
 - 1(1) Caller does not have authority to change mode of specified file
 - 2(2) Security label failure
- 34(22) CHOWN**
 - 0(0) File's owner or group owner changed
 - 1(1) Caller does not have authority to change owner or group owner of specified file
 - 2(2) Security label failure
- 35(23) CLEAR SETID BITS FOR FILE**
 - 0(0) S_ISUID, S_ISGID, and S_ISVTX bits changed to zero (write)
 - No failure cases
- 36(24) EXEC WITH SETUID/SETGID**
 - 0(0) Successful change of UIDs and GIDs
 - No failure cases
- 37(25) GETPSENT**
 - 0(0) Access allowed
 - 1(1) Not authorized to access specified process
- 38(26) INITIALIZE z/OS UNIX PROCESS (DUB)**
 - 0(0) z/OS UNIX process successfully initiated
 - 1(1) User not defined as a z/OS UNIX user (no user profile or no OMVS segment)
 - 2(2) User incompletely defined as a z/OS UNIX user (no UID in user profile)
 - 3(3) User's current group has no GID
- 39(27) z/OS UNIX PROCESS COMPLETION (UNDUB)**
 - 0(0) Process completed
 - No failure cases

- 40(28) KILL**
 - 0(0) Access allowed
 - 1(1) Not authorized to access specified process
 - 2(2) Security label failure
- 41(29) LINK**
 - 0(0) New link created
 - * Failures logged as directory search or check access event types
- 42(2A) MKDIR**
 - 0(0) Directory successfully created
 - * Failures logged as directory search or check access event types
- 43(2B) MKNOD**
 - 0(0) Successful creation of a node
 - * Failures logged as directory search or check access event types
- 44(2C) MOUNT FILE SYSTEM**
 - 0(0) Successful mount
 - * Failures logged as ck_priv event type
- 45(2D) OPEN (NEW FILE)**
 - 0(0) File successfully created
 - * Failures logged as directory search or check access event types
- 46(2E) PTRACE**
 - 0(0) Access allowed
 - 1(1) Not authorized to access specified process
 - 2(2) Security label failure
- 47(2F) RENAME**
 - 0(0) Rename successful
 - * Failures logged as directory search or check access event types
- 48(30) RMDIR**
 - 0(0) Successful rmdir
 - * Failures logged as directory search or check access event types
- 49(31) SETEGID**
 - 0(0) Successful change of effective GID
 - 1(1) Not authorized to setegid
- 50(32) SETEUID**
 - 0(0) Successful change of effective UID
 - 1(1) Not authorized to seteuid
- 51(33) SETGID**
 - 0(0) Successful change of GIDs

- 1(1) Not authorized to setgid
- 52(34) SETUID**
 - 0(0) Successful change of UIDs
 - 1(1) Not authorized to setuid
- 53(35) SYMLINK**
 - 0(0) Successful symlink
 - * Failures logged as directory search or check access event types
- 54(36) UNLINK**
 - 0(0) Successful unlink
 - * Failures logged as directory search or check access event types
- 55(37) UNMOUNT FILE SYSTEM**
 - 0(0) Successful unmount
 - * Failures logged as ck_priv event type
- 56(38) CHECK FILE OWNER**
 - 0(0) User is the owner
 - 1(1) User is not the owner
 - 2(2) Security label failure
- 57(39) CK_PRIV**
 - 0(0) User is authorized
 - 1(1) User not authorized to use requested function
- 58(3A) OPEN SLAVE TTY**
 - 0(0) Access allowed
 - 1(1) Not authorized to access specified process

Event 59(3B): RACLINK EVENT TYPES

The explanations of the event code qualifiers for Event 59 are:

- 0(0) No violation detected
- 1(1) Insufficient authority
- 2(2) Keyword violation detected
- 3(3) Association already defined
- 4(4) Association already approved
- 5(5) Association does not match
- 6(6) Association does not exist
- 7(7) Invalid password or revoked user ID

Event 60(3C)–62(3E): z/OS UNIX XPG4 EVENT TYPES

- 60(3C) CHECK IPC ACCESS**
 - 0(0) Access allowed
 - 1(1) Caller does not have requested access authority

2(2) Security label failure

61(3D) MAKE ISP

0(0) Successful creation of ISP

1(1) Security label failure

62(3E) R_IPC CONTROL

0(0) Access allowed

1(1) Caller does not have requested access authority

2(2) Security label failure

Event 63(3F): z/OS UNIX SETGROUPS EVENT TYPE

0(0) Successful

1(1) Not authorized

Event 64(40): X/OPEN SINGLE UNIX SPECIFICATION EVENT TYPES

64(40) CHECK OWNER TWO FILES

0(0) User is the owner

1(1) User is not the owner

2(2) Security label failure

Event 65(41): z/OS UNIX PASSING OF ACCESS RIGHTS EVENT TYPES

65(41) R_AUDIT

0(0) Successful r_audit

No failure cases

Event 66(42)–67(43): CERTIFICATE EVENT TYPES

66(42) RACDCERT

0(0) No violation detected

1(1) Insufficient authority

67(43) initACEE

0(0) Successful certificate registration

1(1) Successful certificate deregistration

2(2) Insufficient authority to register a certificate

3(3) Insufficient authority to deregister a certificate

4(4) No user ID found for certificate

5(5) Certificate is not trusted

6(6) Successful CERTAUTH certificate registration

7(7) Insufficient authority to register the CERTAUTH certificate

- 8(8) Client security label not equivalent to server's
- 9(9) Invalid use of reserved user ID
- 10(A) No RACF userID found for distributed identity

Event 68(44): GRANT OF INITIAL KERBEROS TICKET

68(44) Kerberos

- 0(0) Success
- 1(1) Failure

Event 69(45): R_PKIServ GENCERT

69(45) RPKIGENC

- 0(0) Successful GENCERT request
- 1(1) Insufficient authority for GENCERT
- 2(2) Successful REQCERT request
- 3(3) Insufficient authority for REQCERT
- 4(4) Successful GENRENEW request
- 5(5) Insufficient authority for GENRENEW
- 6(6) Successful REQRENEW request
- 7(7) Insufficient authority for REQRENEW
- 8(8) Successful PREREGISTER request
- 9(9) Insufficient authority for PREREGISTER

Event 70(46): R_PKIServ EXPORT

70(46) RPKIEXPT

- 0(0) Successful certificate EXPORT request
- 1(1) Unsuccessful certificate EXPORT request due to insufficient authority
- 2(2) Incorrect pass phrase specified for EXPORT

Event 71(47): POLICY DIRECTOR ACCESS CONTROL DECISION

71(47) PDACCESS

This event is reserved for use by Policy Director Authorization Services.

- 0(0) Authorized
- 1(1) Not authorized but permitted because of warning mode
- 2(2) Not authorized due to insufficient traverse authority but permitted because of warning mode
- 3(3) Not authorized due to time-of-day check but permitted because of warning mode
- 4(4) Not authorized

5(5) Not authorized due to insufficient traverse authority

6(6) Not authorized due to time-of-day check

Event 72(48): R_PKIServ QUERY

72(48) RPKIREAD

0(0) Successful admin QUERY or DETAILS request

1(1) Insufficient authority for admin QUERY or DETAILS

2(2) Successful VERIFY request

3(3) Insufficient authority for VERIFY

4(4) Incorrect VERIFY certificate, no record found for this certificate

Event 73(49): R_PKIServ UPDATEREQ

73(49) RPKIUPDR

0(0) Successful admin UPDATEREQ request

1(1) Insufficient authority for admin UPDATEREQ

Event 74(4A): R_PKIServ UPDATECERT

74(4A) RPKIUPDC

0(0) Successful admin UPDATECERT request

1(1) Insufficient authority for admin UPDATECERT

2(2) Successful REVOKE request

3(3) Insufficient authority for REVOKE

Event 75(4B): CHANGE FILE ACL

75(4B) SETFACL

0(0) ACL entry added, changed, or deleted

1(1) Caller does not have authority to change ACL of specified file

2(2) Security label failure

Event 76(4C): REMOVE FILE ACL

76(4C) DELFACL

0(0) Entire ACL deleted

1(1) Caller does not have authority to remove ACL of specified file

2(2) Security label failure

Event 77(4D): SET FILE SECURITY LABEL

77(4D) SETFSECL

0(0) Security label change

1(1) Not authorized to change security label

Event 78(4E): SET WRITE-DOWN PRIVILEGE

78(4E) WRITEDWN

- 0(0) Requested function successful
- 1(1) Not authorized to IRR.WRITEDOWN.BYUSER

Event 79(4F): CRL PUBLICATION

79(4F) PKIDPUBR

- 0(0) Successful publication of revocation information

Event 80(50): R_PKIServ RESPOND

80(50) RPKIRESP

- 0 Successful RESPOND request
- 1 Insufficient authority for RESPOND

Event 81(51): PassTicket Evaluation

81

- 0 Successful request
- 1 Request failed

Event 82(52): PassTicket Generation

82

- 0 Successful generation
- 1 Generation request failed

Event 83(53): R_PKIServ SCEPREQ

83(53) RPKISCEP

- 0 Successful AutoApprove PKCSReq request
- 1 Successful AdminApprove PKCSReq request
- 2 Successful GetCertInitial request
- 3 Rejected PKCSReq or GetCertInitial request
- 4 Incorrect SCEP transaction ID specified for GetCertInitial
- 5 Insufficient authority for SCEPREQ

Event 84(54): R_Datalib RDATAUPD

84(54) RDATAUPD

- 0 Successful NewRing
- 1 Not authorized to call NewRing
- 2 Successful DataPut
- 3 Not authorized to call DataPut
- 4 Successful DataRemove

- 5 Not authorized to call DataRemove
- 6 Successful DelRing
- 7 Not authorized to call DelRing

Event 85(55): PKIAURNW

85(55) PKIAURNW

- 0 Successful autoRenew

Event 86(56): R_PgmSignVer

86(56) R_PgmSignVer

- 0 Successful signature verification
- 1 Signature appears valid but root CA certificate not trusted
- 2 Module signature failed verification
- 3 Module certificate chain incorrect
- 4 Signature required but module not signed
- 5 Signature required but signature has been removed
- 6 Program verification module not loaded. Program verification was not available when attempt was made to load this program.
- 7 The Algorithmic self test failed while verifying the program verification module.

Event 87(57): RACMAP

87(57) RACMAP

- 0(0) No violation detected
- 1(1) Insufficient authority (no update to RACF database)

Event 88(58): AUTOPROF

88(58) AUTOPROF

- 0 Successful profile modification

Event 89(59): RPKIQREC

89(59) RPKIQREC

- 0 Successful user QRECOVER request
- 1 Insufficient authority for user QRECOVER

Audit function codes for z/OS UNIX System Services

Table 9. Audit Function Codes for z/OS UNIX System Services

Code	Name	Description
1	AFC_ACCESS	check file accessibility
2	AFC_CHAUDIT_U	change user audit options
3	AFC_CHDIR	change current working directory
4	AFC_CHMOD	change file modes

Table 9. Audit Function Codes for z/OS UNIX System Services (continued)

Code	Name	Description
5	AFC_CHOWN	change owner and group of file
6	AFC_DUB	initialize a process
7	AFC_EXEC	execute with new jobname
8	AFC_FCHAUDIT_U	change user audit options when file is open
9	AFC_FCHMOD	change file modes when file is open
10	AFC_FCHOWN	change owner and group of file when open
11	AFC_GETCWD	get current working directory
12	AFC_GETPSENT	get process entry
13	AFC_KILL	signal a process
14	AFC_LINK	link to a file
15	AFC_LSTAT	get file status - do not resolve ending symlink
16	AFC_MKDIR	make a directory
17	AFC_MKNOD	make a file node
18	AFC_MOUNT	mount a file system (nosetuid)
19	AFC_OPEN	open a file
20	AFC_OPENDIR	open a directory
21	AFC_PATHCONF	get configurable path name variables
22	AFC_PTRACE	debug a process
23	AFC_READLINK	read a symbolic link
24	AFC_RENAME	rename a file
25	AFC_RMDIR	remove a directory
26	AFC_SETEGID	set effective GID
27	AFC_SETEUID	set effective UID
28	AFC_SETGID	set real and/or effective GID
29	AFC_SETUID	set real and/or effective UID
30	AFC_STAT	get file status
31	AFC_SYMLINK	create a symbolic link
32	AFC_UNLINK	remove directory entries (delete a file)
33	AFC_UNMOUNT	unmount a file system (nosetuid)
34	AFC_UTIME	set file access/modification times
35	AFC_UNDUB_EXIT	terminate a process
36	AFC_WRITE	write to a file (clear setid bits)
37	AFC_CHAUDIT_A	change auditor audit options
38	AFC_FCHAUDIT_A	change auditor audit options when file is open
39	AFC_LOOKUP	path name resolution
40	AFC_TTYNAME	get pathname of terminal
41	AFC_IOCTL	get path name
42	AFC_GETMNT	get mount entry
43	AFC_QUIESCE	quiesce a file system (nosetuid)
44	AFC_UNQUIESCE	unquiesce a file system (nosetuid)
45	AFC_VREGISTER	server registration, v_reg
46	AFC_VRESOLVEPN	server resolve pathname, v_rpn
47	AFC_VLOOKUP	server lookup, v_lookup
48	AFC_VREADWRITE	server read write, v_rdwr
49	AFC_VREADDIR	server read directory, v_readdir
50	AFC_SIGACTION	change Osigset action
51	AFC_VCREATE	server create, v_create
52	AFC_VMAKEDIR	server make directory, v_mkdir
53	AFC_VSYMLINK	server symbolic link, v_symlink
54	AFC_VSETATTR	server file attributes, v_setattr
55	AFC_VLINK	server link, v_link
56	AFC_VREMOVEDIR	server remove directory, v_rmdir

Table 9. Audit Function Codes for z/OS UNIX System Services (continued)

Code	Name	Description
57	AFC_VREMOVE	server remove, v_remove
58	AFC_VRENAME	server rename, v_rename
59	AFC_CHATTR	change file attributes
60	AFC_FCHATTR	change file attributes when file is open
61	AFC_THLMT	set thread limit
62	AFC_MSGCTL	message control
63	AFC_MSGGET	get message queue
64	AFC_MSGRCV	message receive
65	AFC_MSGSND	message send
66	AFC_SEMCTL	semaphore control
67	AFC_SEMGET	get set of semaphores
68	AFC_SEMOP	semaphore operations
69	AFC_SHMAT	shared memory attach
70	AFC_SHMCTL	shared memory control
71	AFC_SETREUID	set real and/or effective UID
72	AFC_SHMGET	shared memory get
73	AFC_WGETIPC	query IPC status
74	AFC_REMOVE	remove
75	AFC_SET_MODE	set mode
76	AFC_SET_MSGQB	set max bytes for msg queue
77	AFC_SET_GID	set supplementary groups
78	AFC_PASSWORD	verify password
79	AFC_LCHOWN	change owner and group of a symbolic link
80	AFC_TRUNCATE	truncate a file
81	AFC_PFSCTL	control function for the physical file system
82	AFC_SETRLIMIT	set maximum resource consumption
83	AFC_SETPRIORITY	set process scheduling priority
84	AFC_NICE	change priority of a process
85	AFC_SETREGID	set real and effective GID
86	AFC_WRITEV	write on a file
87	AFC_FCHDIR	change working directory
88	AFC_CHROOT	change root directory
89	AFC_REALPATH	resolve path name
90	AFC_STATVFS	get file system information
91	AFC_BIND	bind a name to a socket
92	AFC_SOCKET	create an endpoint for communication
93	AFC_THREAD_SEC	thread level security
94	AFC_AUTHCHECK	authority check
95	AFC_ACC_SEND	send access rights
96	AFC_ACC_RECV	receive access rights
97	AFC_ACC_DISC	discard access rights
98	AFC_NEWGRP	newgrp shell utility
99	AFC_CONSOLE	console communication service
100	AFC_SERV_INIT	WLM service
101	AFC_SPAWN	spawn with user ID
102	AFC_SWAP_SERV	swap services
103	AFC_WLMC	WLM C and C++
104	AFC_LOGIN	__login system call
105	AFC_MOUNT_SETUID	mount a file system (setuid)
106	AFC_UNMOUNT_SETUID	unmount a file system (setuid)
107	AFC_QUIESCE_SETUID	quiesce a file system (setuid)
108	AFC_UNQUIESCE_SETUID	unquiesce a file system (setuid)
109	AFC_CHMOUNT	change mount point of a file system
110	AFC_CHMOUNT_SETUID	change mount point of a file system (setuid)

Table 9. Audit Function Codes for z/OS UNIX System Services (continued)

Code	Name	Description
111	AFC_SETFACL	add, alter, or delete an access control list
112	AFC_SHUTDOWN_REG	shutdown registration
113	AFC_EACCESS	check effective access
114	AFC_SETFSECL	Set file security label
115	AFC_POE	Provide port of entry identifier
116	AFC_LCHATTR	Change file attributes
117	AFC_UNAVAILABLE	AFC unavailable
118	AFC_MOUNT_NA	mount no audit
119	AFC_MOUNT_U	mount
120	AFC_MOUNT_UNA	user mount no audit
121	AFC_UNMOUNT_U	user unmount
122	AFC_UNMOUNT_UNA	user unmount no audit
123	AFC_FSACCESS	File system access
124	AFC_SHMMCV	Shared memory segment mutexes and condition variables
125	AFC_ENDOF_TAB	end of table

Chapter 4. The data security monitor (DSMON)

RACF enables you to protect resources, but the protection is only as good as the implementation. You need a way to verify that the security mechanisms actually in effect are the ones intended. DSMON helps provide this information.

DSMON is a program that produces reports on the status of the security environment at your installation and, in particular, on the status of resources that RACF controls. You can use the reports to audit the current status of your installation's system security environment by comparing the actual system characteristics and resource protection levels with the intended characteristics and levels. You can also control the reporting that DSMON does by specifying control statements that request certain functions for user input.

The DSMON program

The data security monitor (DSMON) is a program that normally runs while RACF is active.

If the installation has not defined ICHDSM00 (DSMON) as a controlled program, you must have the AUDITOR attribute to run DSMON.

If DSMON has been defined as a controlled program, you must have at least EXECUTE or READ access to the ICHDSM00 resource in the PROGRAM class. Additionally, with READ access using the PROGRAM class if DSMON is not running in a clean program-controlled environment, you also need the AUDITOR attribute.

You can specify DSMON control statements to produce the reports you want and control the number of lines per page for each report. The output from DSMON consists of a message data set and an output data set for the reports.

Note:

1. To find out if DSMON is a controlled program at your MVS installation, contact your RACF security administrator.
2. If your installation has a RACF database that is shared by MVS and z/VM and you want to obtain reports for both systems, you must run DSMON on the MVS system.
3. If you run DSMON while RACF is inactive, DSMON produces only the system report.

How to run DSMON

DSMON runs as an authorized program facility (APF)-authorized batch program. DSMON can also be run on TSO if IKJTSO00 is configured correctly; it can reside in any PARMLIB data set.

To invoke DSMON, you can use the sample job control language (JCL) statements in Figure 15 on page 96. A SYSIN DD statement lets you specify DSMON control statements that can perform selected DSMON functions for specified user input. The words that appear in lowercase are parameters that you can change.

```

//stepname EXEC PGM=ICHDSM00
//SYSPRINT DD SYSOUT=A
//SYSUT2 DD SYSOUT=A
//SYSIN DD *
LINECOUNT 55
FUNCTION all
USEROPT USRDSN sivle.memo.text

```

Figure 15. Specifying DSMON JCL

SYSPRINT

Defines the sequential message data set (for example, SYSOUT) for status and error messages. SYSPRINT has a variable block (VB) format; block size, if specified, must be 137 (LRECL of 133 plus 4 for the block length) or greater.

SYSUT2

Defines the output listing data set (for example, SYSOUT) for the printed reports that DSMON generates. SYSUT2 has a fixed block (FB) format; block size, if specified, must be a multiple of 133.

SYSIN

Defines the control data set that contains DSMON control statements. SYSIN is required if you want to select specific DSMON functions. The control data set can be one of the following:

- A data set defined as in-stream data
- A data set defined as a sequential data set
- A data set defined as a member of a partitioned data set

Block size, if specified, must be a multiple of 80.

If you do not specify SYSIN, all DSMON functions except USRDSN are performed. (The USRDSN function requires you to specify a list of user data sets on the USEROPT control statement.)

DSMON control statements

The three DSMON control statements that allow you to control DSMON reporting are:

- LINECOUNT
- FUNCTION
- USEROPT

Define these statements as part of the SYSIN DD statement in the JCL (see Figure 15).

Entering DSMON control statements

DSMON control statements can be entered in any order, one per input line, using columns 1 through 72. You can enter uppercase or lowercase characters. Use commas or blanks to separate list items in each DSMON statement.

You can include comments by entering a /* beginning in column 1. If you want to continue a control statement on a following line, break the statement at any place a blank or comma is allowed and insert a blank followed by a trailing hyphen (-) before you continue to the next line. For example:

```

/* Start of user data sets
USEROPT USRDSN jim.memo.text vol=8V0L03 -
jim.report.script

```


The DSMON control statements are:

LINECOUNT number

specifies the number of lines per page for reports. The valid values for number are 0 or a number in the range of 40 through 99. A value of 0 indicates that a page break occurs only at the start of a new report. If you do not specify LINECOUNT, the default is 55 lines per page. If you specify more than one LINECOUNT statement, RACF uses only the last one.

Note: The LINECOUNT statement controls the number of lines per page for the SYSUT2 data set. It does not affect the number of lines per page for the SYSPRINT message data set, fixed at 55 lines per page.

FUNCTION function-name

specifies the DSMON function or functions you want to include.

The default is ALL, which causes DSMON to generate all reports except USRDSN. For a complete description of the DSMON reports specified for function-name, see “Functions DSMON uses” on page 98.

USEROPT function-name user-input

defines user input to be processed by the function you specify. Function-name specifies the function to process the user-input; user-input specifies the actual input you want processed. The valid functions you can specify for function-name on the USEROPT control statement are:

- USRDSN
- RACGRP

Be sure to use one USEROPT control statement for each valid function you want to process the specified input.

USEROPT control statement

USEROPT and USRDSN

Specifying USRDSN with USEROPT causes DSMON to list the RACF protected status of the selected user data set or sets. To obtain information processed by USRDSN, specify USEROPT followed by one or more blanks, then followed by USRDSN and the data set name and volume or both for which you want information.

For example, if you want to specify a cataloged data set, use the full data set name after USRDSN:

```
USEROPT USRDSN jim.memo.text
```

If you want to specify an uncataloged data set, use the full data set name and volume:

```
USEROPT USRDSN jim.memo.text VOL=volser
```

You can use the USRDSN option with other DSMON functions. For example, the following specifies that all other functions in addition to USRDSN are to be performed:

```
FUNCTION ALL  
USEROPT USRDSN jim.memo.text VOL=volser
```

Note that FUNCTION ALL is the default; if you omit it, DSMON produces all reports. The following specifies that only the USRDSN function is to be performed on the specified data set:

```
FUNCTION USRDSN
USEROPT USRDSN jim.memo.text
```

In the next example, USRDSN is specified for a list of data sets:

```
FUNCTION USRDSN
USEROPT USRDSN jim.memo.text -
          VOL=8VOL03 jim.test.obj -
          jim.racf.cntl jim.racf.clist
```

Note: The VOL keyword does not apply for SMS.

USEROPT and RACGRP

Specifying RACGRP with USEROPT causes DSMON to list the group tree and its levels for any specified RACF group name. The following specifies RACGRP for FUNCTION and the RACF group "payroll" (for which all subordinate groups are to be retrieved) for USEROPT RACGRP:

```
FUNCTION RACGRP
USEROPT RACGRP payroll
```

If you specify SYS1 for USEROPT RACGRP, DSMON lists all group names in the system. If you want all DSMON reports but do not specify USEROPT RACGRP, SYS1 is the default group name for the RACF group tree report. You can specify any RACF-defined group. For more information about the DSMON report RACGRP produces, see "Group tree report" on page 102.

USEROPT considerations

A JCL REGION= keyword may limit the number of USEROPT control statements that can be specified. If many USEROPT statements are specified, increase the REGION= keyword value accordingly. Users may also run a multistep job if increasing the region size is unsuccessful.

Functions DSMON uses

DSMON generates different kinds of reports that you can specify on the FUNCTION or USEROPT control statements. After completing each function on the control statement (except for the system report), DSMON issues a message to SYSPRINT stating whether the report executed successfully or unsuccessfully.

If the report ended unsuccessfully, DSMON issues an error code that indicates the cause of the failure. In most cases, DSMON continues processing with the next control statement.

Table 10 on page 99 summarizes the DSMON reports that are generated when you use the FUNCTION control statement. Table 11 on page 99 summarizes the DSMON reports that are generated when you use the USEROPT control statement. You can specify the kind of report you want by modifying function name on each control statement. Both tables list the type of report produced and the information (or checks) each report provides.

Table 10. Reports Specified by the FUNCTION Control Statement

Function-name	Type of Report	Information Provided
SYSTEM	System Report	<ol style="list-style-type: none"> 1. Identification number of the processor complex 2. Model number of the processor complex 3. Name, version, and release number of the operating system 4. System residence volume 5. System identifier used by the System Management Facilities 6. RACF version and release number and whether RACF is active
RACGRP	Group Tree Report (also used with USEROPT; Table 11)	Group name and level in hierarchy for entire system
SYSPT	Program Properties Table Report	All information (see sample report)
RACAUT	RACF Authorized Caller Table Report	All information (see sample report)
RACCDT	RACF Class Descriptor Table Report	All information (see sample report)
RACEXT	RACF Exits Report	All information (see sample report)
RACGAC	RACF Global Access Table Report	All information (see sample report)
RACSPT	RACF Started Procedures Table Report	All information (see sample report)
RACUSR	Selected User Attribute Report and Selected User Attribute Summary Report	All information (see sample reports)
SYSLNK	Selected Data Sets Report	All LNKSTxx data set members of the SYS1.PARMLIB library
SYSAPF	Current Link List Data Set Report	Authorized program facility (APF) libraries
SYSCAT	Selected Data Sets Report	<p>Master catalog and all user catalogs. Requires additional authorization to obtain information on user catalogs</p> <p>Note: If you have a FACILITY class profile that protects SYSCAT resource ICHDSM00.SYSCAT and you do not have READ access, DSMON suppresses the user catalog listing and issues message ICH66134I, notifying you of the insufficient authorities</p>
RACDST	Selected Data Sets Report	Primary and backup RACF databases
SYSDDS	Selected Data Sets Report	Selected system data sets
USRDSN	Selected Data Sets Report (used with USEROPT; Table 11)	Selected user data sets

Table 11. Reports Specified by the USEROPT Control Statement

Function-name	Type of Report	Information Provided
USRDSN	Selected User Data Sets Report	Selected user data sets
RACGRP	Group Tree Report	Group name and level in hierarchy for user-specified group

DSMON reports

DSMON produces the following reports:

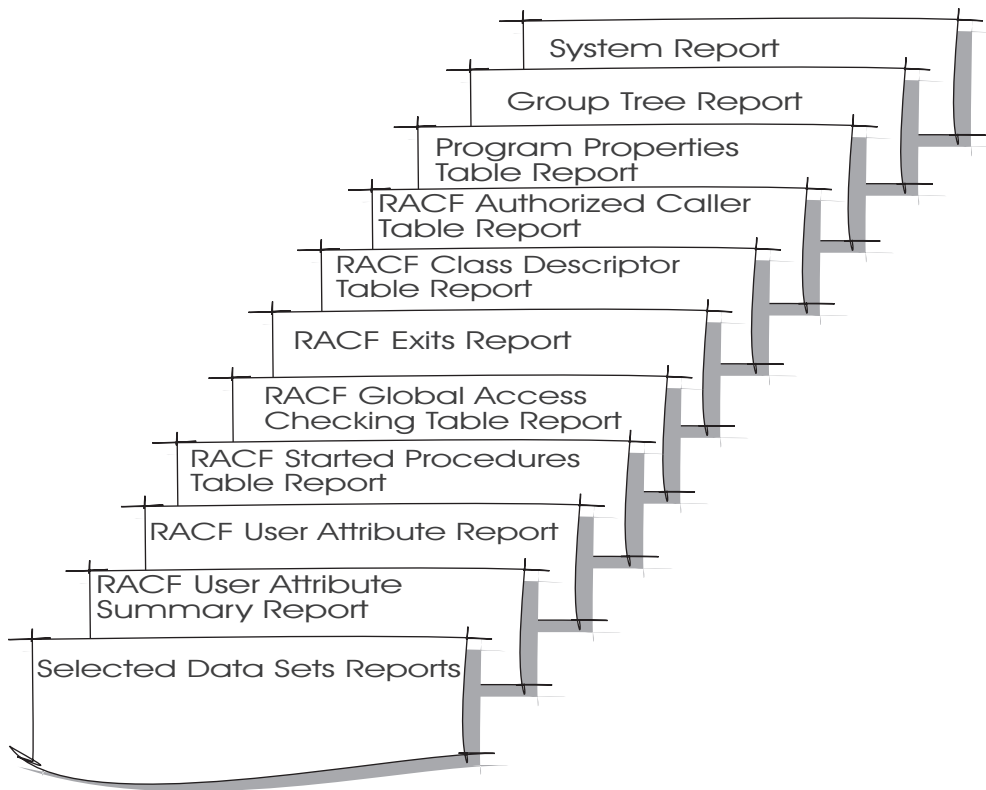


Figure 16. Reports produced by DSMON

Note: Producing the group tree report or the selected user attribute report and selected user attribute summary report can have an impact on system performance. Depending on the size of and load on your RACF databases, you should consider running these DSMON reports during slack time.

The information in the DSMON reports answers many of your audit questions. (See “Conducting the audit” on page 9.)

System report

The system report contains:

- The identification number and model of the processor complex
- The name, version, and release of the operating system
- The serial number of the system residence volume
- The system identifier (SMF-ID) that SMF uses

The report also specifies the RACF version and release number and whether RACF is active. If RACF is inactive, either because it was not activated at IPL or because it has been deactivated by the RVARY command, DSMON prints a message.

You can use the system report to verify that the system has the expected hardware and software. In addition, you can verify the status of RACF.

Note: DSMON always produces the system report. However, if RACF is not installed and active, DSMON produces only the system report and then stops.

Column headings

The report contains the following information:

CPU-ID

is the identification number of the processor complex on which the system is running.

CPU MODEL

is the model number of the processor complex.

OPERATING SYSTEM/LEVEL

specifies the name, version and release of the operating system, the product FMID for the operating system, and the installation's personalized name, if the information is present in the communications vector table (CVT).

SYSTEM RESIDENCE VOLUME

specifies the serial number of the volume on which the system resides.

SMF-ID

is the system identifier that the system management facilities (SMF) uses when creating log records.

Report messages

The following messages may appear at the end of the report:

RACF FMID HRFnnnn IS ACTIVE

Explanation: The specified FMID of RACF was not activated during initial program load (IPL).

Note: Under normal circumstances, this message should not appear. If it does, notify your RACF security administrator or your installation manager.

RACF FMID HRFnnnn IS INACTIVE

Explanation: The specified FMID of RACF is active. In most cases, this is the message that appears on the report.

RACF FMID HRFnnnn HAS BEEN DEACTIVATED

Explanation: The specified FMID of RACF has been deactivated by the RVARY command; this situation is normally temporary.

RACF IS NOT INSTALLED

Explanation: DSMON cannot locate the RACF communications vector table (RCVT), indicating that RACF has not been installed.

Note: Under normal circumstances, this message should not appear. If it does, notify your RACF security administrator or your installation manager.

S Y S T E M R E P O R T

```

CPU-ID                111606
CPU MODEL             2064
OPERATING SYSTEM/LEVEL z/OS 1.2.0      HBB7705   Test System 2390
SYSTEM RESIDENCE VOLUME DR250B
SMF-ID               IM13
RACF FMID HRF7705 IS ACTIVE
  
```

Figure 17. Sample System Report

Group tree report

The group tree report lists all subgroups for the SYS1 group and continues to list subgroups for those subgroups on down the group tree. Alternately, if a user-specified group name is specified for the USEROPT control statement, the report lists all subgroups for that user-supplied group. The report provides the owner's name for each group, if the owner is not the superior group.

You can use the group tree report to examine the overall RACF group structure for your system. You can also determine how the group related attributes (group-SPECIAL, group OPERATIONS, and group AUDITOR) for users associated with each subgroup are related. In this way, you can decide whether the group authorities are structured effectively for your system.

Column Headings

LEVEL

Starting with the highest requested group, the group level number that indicates the relative nesting level of the group or subgroup within the requested group tree. SYS1 is always 1; the groups with SYS1 as their superior group are 2, and so on down the group tree.

GROUP

is the name of the RACF-defined group.

(OWNER)

is the name of the owner of the group. This name is listed only if the owner is not the superior group.

Report Messages

An arrow (==>) in the report indicates that the information has overflowed the right margin. The missing information appears after the main body of the report is printed. The characters -----CONTINUATION----- appear before the overflowed information, and the discontinued level number, group, and owner name (if the name is not the same as that of the superior group) appear in the left margin.

```

                                R A C F   G R O U P   T R E E   R E P O R T
-----
LEVEL  GROUP      (OWNER)
-----
  1   SYS1        (IBMUSER )
  2   | SYSPROG   (IBMUSER )
  2   | RACFADMN (IBMUSER )
  
```

Figure 18. Sample Group Tree Report

Program properties table report

The program properties table report lists all the programs in the program properties table (PPT). The report also indicates whether each program is authorized to bypass password protection and whether it runs in a system key. The programs shown in this report may be able to bypass password protection for password protected data sets and thus also bypass all RACF protection for RACF-protected resources.

You can use the program properties table report to verify that only those programs that should be authorized to bypass password protection are, in fact, able to do so. Such programs are normally communication and database control programs, or other system control programs. You can also verify that only those programs that must run in a system key are authorized to do so.

Column Headings

PROGRAM NAME

is the name of the program, as defined in the PPT.

BYPASS PASSWORD PROTECTION

indicates whether the program is authorized to bypass password protection checking when accessing RACF-protected or password-protected data sets. The value is either YES or NO.

SYSTEM KEY

indicates whether the program is authorized to run in a system key (keys 0-7) and is thus able to bypass system security controls. The value is either YES or NO.

Report Messages

The following message may appear below the report column headings:

NO ENTRIES IN PROGRAM PROPERTIES TABLE

Explanation: There are no entries in the program properties table. This message indicates an abnormal condition because the program properties table should contain several entries that were supplied by IBM.

PROGRAM PROPERTIES TABLE REPORT		
PROGRAM NAME	BYPASS PASSWORD PROTECTION	SYSTEM KEY
IEDQTCAM	NO	YES
ISTINM01	YES	YES
IKTCAS00	NO	YES
AHLGTF	NO	YES
HHLGTF	NO	YES
IHLGTF	NO	YES
IEFIIC	NO	YES
IEEMB860	YES	YES
IEEVMNT2	NO	YES
IASXWR00	NO	YES
CSVVFCRE	NO	YES
HASJES20	YES	YES
DFSMVRC0	NO	YES
IATINTK	YES	YES
DXRRLM00	NO	YES
APSPPIEP	NO	YES
AKPCSI EP	NO	YES
IATINTKF	YES	YES
DSNYASCP	NO	YES
DSNUTILB	NO	YES
IEAVTDSV	YES	YES
IFASMF	NO	YES
CSVLLCRE	YES	YES
AVFMNBLD	NO	YES
ERBMFMFC	NO	NO
ERB3GMFC	NO	NO
IGGOCLX0	NO	YES
IGDSSI01	YES	YES
COFMINIT	YES	YES
COFMISD0	NO	YES

Figure 19. Sample Program Properties Table Report

RACF authorized caller table report

The RACF authorized caller table report lists the names of all programs in the RACF authorized caller table. The report also indicates whether each program is authorized to issue a VERIFY (RACINIT) request (which performs user verification) or a LIST (RACLIST) request (which loads profiles into main storage), or both.

You can use this report to verify that only those programs authorized to modify an access control environment element (ACEE) are able to issue a VERIFY request. This verification is an important security requirement because the ACEE contains a description of the current user. This description includes the user ID, the current connect group, the user attributes, and the group authorities. A program that is authorized to issue a VERIFY request can alter the ACEE to simulate any user ID.

You can also use the report to verify that only those programs authorized to access any profile on the RACF data set are able to issue a LIST request. Because profiles contain complete descriptions of the characteristics associated with RACF-defined entities, you must carefully control access to them.

Note: IBM does not recommend using the RACF authorized caller table.

Column Headings

MODULE NAME

is the name of the program module as it is defined in the RACF authorized caller table.

RACINIT AUTHORIZED

indicates whether the module is authorized to issue a VERIFY request. The value is either YES or NO.

RACLIST AUTHORIZED

indicates whether the module is authorized to issue a LIST request. The value is either YES or NO.

Report Messages

The following message may appear below the report column headings:

NO ENTRIES IN RACF AUTHORIZED CALLER TABLE

Explanation: There are no entries in the RACF authorized caller table. This message does not indicate an error condition. When RACF is initially installed, for example, the RACF authorized caller table normally contains no entries.

MODULE NAME	RACF RACINIT AUTHORIZED	AUTHORIZED RACLIST AUTHORIZED	CALLER TABLE	REPORT

NO ENTRIES IN RACF AUTHORIZED CALLER TABLE				

Figure 20. Sample RACF Authorized Caller Table Report

RACF class descriptor table report

The class descriptor table report lists class name and status for all general resource classes in the class descriptor table, including information about auditing activity, statistics, the activity of OPERATIONS users, and the universal access authority (UACC).

You can use the class descriptor table report to determine the resource classes defined to RACF for your system. In this way, you can obtain information about the protection status of any resource in the class descriptor table.

Column Headings

CLASS NAME

is the class name found in the RACF class descriptor table. The dynamic classes are noted with a "(D)" after the class name.

STATUS

indicates whether the class is active or inactive.

AUDITING

indicates whether there is auditing for the class. The value is either YES or NO.

STATISTICS

indicates whether RACF is gathering statistics for the class. The value is either YES or NO.

DEFAULT UACC

indicates that the default UACC defined for the class in the class descriptor

table. RACF uses this UACC for profiles defined to the class, unless the UACC operand is specified on the RDEFINE command that writes the profile.

The following values may appear:

- ALTER
 - For discrete profiles, ALTER indicates that, by default, all users have control over the resource and the resource profile and can authorize other users or groups (or both) to access the resource.
 - For generic profiles, ALTER indicates that, by default, all users have control over the resource and can allocate data sets protected by the generic profile. Only the profile owner has full control over the resource profile.
- CONTROL indicates that, by default, all users have access authority to update, insert, or delete records in the VSAM data set and perform other operations as if the data set password were supplied.
- UPDATE indicates that, by default, all users can access the resource for both reading and writing.
- READ indicates that, by default, all users can access the resource for reading only.
- NONE indicates that, by default, users cannot access the resource.
- ACEE indicates that the UACC is taken from the accessor environment element (ACEE).

OPERATIONS

indicates whether RACF is to use the OPERATIONS attribute authority during authorization checking. A value of YES indicates RACF performs authorization checking; a value of NO indicates it does not.

Report Messages

The following message may appear below the report column headings:

NO ENTRIES IN THE RACF CLASS DESCRIPTOR TABLE

Explanation: There are no entries in the class descriptor table. RACF includes a basic class descriptor table, which is required for RACF processing. If you receive this message, report the condition to your RACF security administrator or installation manager.

CLASS NAME	RACF STATUS	CLASS AUDITING	DESCRIPTOR STATISTICS	TABLE DEFAULT UACC	REPORT OPERATIONS ALLOWED
\$CAMP (D)	INACTIVE	NO	NO	NONE	NO
#NUMCLAS (D)	INACTIVE	YES	YES	NONE	NO
@NEWCLAS (D)	INACTIVE	NO	NO	NONE	NO
AIMS	INACTIVE	NO	NO	NONE	NO
APPL	INACTIVE	NO	NO	NONE	NO
DASDVOL	INACTIVE	NO	NO	ACEE	YES
DBCLASS5 (D)	INACTIVE	NO	NO	NONE	NO
DBCLASS6 (D)	INACTIVE	NO	NO	NONE	NO
DSNR	INACTIVE	NO	NO	ACEE	NO
FACILITY	INACTIVE	NO	NO	NONE	NO
GCICSTRN	INACTIVE	NO	NO	NONE	NO
GDASDVOL	INACTIVE	NO	NO	ACEE	YES
GIMS	INACTIVE	NO	NO	NONE	NO
GLOBAL	INACTIVE	NO	NO	NONE	NO
GMBR	INACTIVE	NO	NO	NONE	NO
GTERMINL	INACTIVE	NO	NO	ACEE	NO
PCICSPSB	INACTIVE	NO	NO	NONE	NO
QCICSPSB	INACTIVE	NO	NO	NONE	NO
RACFVARS	INACTIVE	NO	NO	NONE	NO
RVARSMBR	INACTIVE	NO	NO	NONE	NO
SECLABEL	INACTIVE	NO	NO	NONE	NO
TAPEVOL	INACTIVE	NO	NO	ACEE	YES
TCICSTRN	INACTIVE	NO	NO	NONE	NO
TERMINAL	INACTIVE	NO	NO	ACEE	NO
TIMS	INACTIVE	NO	NO	NONE	NO
VMBATCH	INACTIVE	NO	NO	NONE	YES
VMCMD	INACTIVE	NO	NO	NONE	YES
VMDISK	INACTIVE	NO	NO	NONE	YES
VMRDR	INACTIVE	NO	NO	NONE	YES

Note: DSMON generates its RACF Class Descriptor Table report listing classes from both the static and dynamic class descriptor tables. The dynamic classes will be noted with a "(D)" after the class name in the first column of the report. Also, note that the classes are listed in alphabetical order.

Figure 21. Class Descriptor Table Report

RACF exits report

The RACF exits report lists the names of all the installation-defined RACF exit routines and specifies the size of each exit routine module. For RACF's static exits, DSMON prints an error message if the RACF communications vector table (RCVT), which contains the address of each RACF exit routine module, indicates that an exit routine module should exist but the module cannot be loaded, or the entry address does not correspond with the address specified in the RCVT.

You can use this report to verify that the only active exit routines are those that your installation has defined. The existence of any other exit routines may indicate a system security exposure, because RACF exit routines could be used to bypass RACF security checking. Similarly, if the length of an exit routine module differs from the length of the module your installation defined, the module may have unauthorized modifications.

Note: For the dynamic exits IRREVX01 and IRRVAF01, note that these are the exit names, not necessarily the module names associated with the exit. MVS dynamic exit service supports multiple exit routines for a single exit point. The DSMON exits report lists IRREVX01 or IRRVAF01 when at least one active exit routine is

defined at the time the report is created. The report will not include any routine names or sizes, listing the length of IRREVX01 and IRRVAF01 as NA (not available).

See *z/OS MVS Programming: Authorized Assembler Services Reference ALE-DYN* for information on the dynamic exit service CSVDYNEX macro which is used by RACF to define and query its dynamic exits. See *z/OS MVS System Commands* for information on the DISPLAY command's support of dynamic exits. The MVS DISPLAY command can be used to find the names of the modules associated with the exits. Additionally see the exit chapter in *z/OS Security Server RACF System Programmer's Guide* for information on IRREVX01 and IRRVAF01.

Column Headings

EXIT MODULE NAME

is the name of the RACF exit routine module, as defined by your installation.

MODULE LENGTH

is the length of the exit routine module in bytes (decimal).

Report Messages

The following message may appear below the report column headings:

NO RACF EXITS ARE ACTIVE

Explanation: There are no active RACF exit routines. This absence does not indicate an abnormal condition, unless your installation has defined RACF exit routines.

```

                                     R A C F   E X I T S   R E P O R T
EXIT MODULE          MODULE
NAME                 LENGTH
-----
NO RACF EXITS ARE ACTIVE
```

Figure 22. Sample RACF Exits Report

RACF global access checking table report

The global access checking table report lists all entries in the global access checking table. Each entry consists of a resource name and its associated global access checking authority level.

Also, you can use the global access checking table report to determine whether protection for a sensitive resource is adequate. By examining the global access information for an entry, you can discover whether the global access authority level provides the right security for the resource.

Column Headings

CLASS NAME

is the class name found in the global access checking table.

ENTRY NAME

is the entry name or names defined in each class. If the GLOBAL class is inactive, GLOBAL INACTIVE appears in this column. If the GLOBAL class is active but no members are defined for the class, NO ENTRIES appears in the column.

ACCESS LEVEL

specifies the global access checking authority level for the entry.

Report Messages

The following message may appear below the report column headings:

GLOBAL INACTIVE

Explanation: There are no entries in the RACF global access checking table. This message does not indicate an error condition. When RACF is initially installed, for example, the RACF global access checking table normally contains no entries.

CLASS NAME	ACCESS LEVEL	R A C F ENTRY NAME	G L O B A L	A C C E S S	T A B L E	R E P O R T

DATASET		-- GLOBAL	INACTIVE	--		
RVARSMBR		-- GLOBAL	INACTIVE	--		
SECLABEL		-- GLOBAL	INACTIVE	--		
DASDVOL		-- GLOBAL	INACTIVE	--		
TAPEVOL		-- GLOBAL	INACTIVE	--		
TERMINAL		-- GLOBAL	INACTIVE	--		
APPL		-- GLOBAL	INACTIVE	--		
TIMS		-- GLOBAL	INACTIVE	--		
AIMS		-- GLOBAL	INACTIVE	--		
TCICSTRN		-- GLOBAL	INACTIVE	--		
PCICSPSB		-- GLOBAL	INACTIVE	--		
GMBR		-- GLOBAL	INACTIVE	--		
DSNR		-- GLOBAL	INACTIVE	--		
FACILITY		-- GLOBAL	INACTIVE	--		
VMMDISK		-- GLOBAL	INACTIVE	--		
VMRDR		-- GLOBAL	INACTIVE	--		
VMCMD		-- GLOBAL	INACTIVE	--		
VMNODE		-- GLOBAL	INACTIVE	--		
VMATCH		-- GLOBAL	INACTIVE	--		
SCDMBR		-- GLOBAL	INACTIVE	--		
FCICSFCT		-- GLOBAL	INACTIVE	--		
JCICSJCT		-- GLOBAL	INACTIVE	--		
DCICSDCT		-- GLOBAL	INACTIVE	--		
SCICSTST		-- GLOBAL	INACTIVE	--		

Figure 23. Sample RACF Global Access Checking Table Report

RACF started procedures table reports

The status of the STARTED class determines the started procedures table reports that get generated. If the STARTED class is not active, the report is created using the installation replaceable load module, ICHRIN03, as shown in Figure 24 on page 111. If the STARTED class is active, two reports are generated. Along with the report generated for the installation replaceable load module, ICHRIN03, a second report is created using the STARTED class profiles. An example of this second report is shown in Figure 25 on page 112.

The started procedures table report lists each entry in the started procedures table. Each entry contains the procedure name, user identification, the group name associated with the procedure, the privileged status, and the trusted status. If the STARTED class is active, the report that gets generated also shows the job name associated with the procedure and the TRACE attribute.

In order for the started procedures table report to show your installation's currently active profiles, you should issue:

```
SETR RACLIST(STARTED) REFRESH
```

before running the report. Be aware that this command could cause some disruption if profiles are being changed on the system at the exact time the command is issued.

Using STARTED class profiles allows you to dynamically change the table entries without having to re-IPL. For more details, refer to *z/OS Security Server RACF Security Administrator's Guide*.

You can use the started procedures table report to determine which started procedures are defined to RACF and which RACF user IDs and groups they will use. RACF user IDs associated with the started procedure can access RACF-protected resources. Therefore, you can check the information in the RACF started procedures table to determine which users and groups are associated with the started procedure that RACF recognizes, and determine whether those users are privileged or trusted.

You can also use the report to determine which started procedures are privileged or trusted. If the started procedure has the PRIVILEGED attribute, it can bypass all RACROUTE REQUEST=AUTH and REQUEST=FASTAUTH processing, including the security classification checks, and can therefore affect the overall security of the system. TRUSTED means the same as PRIVILEGED, except that auditing can be requested by using the SETROPTS LOGOPTIONS command or the UAUDIT operand on the ALTUSER command.

Column Headings

PROCEDURE NAME

is the procedure name, or an asterisk ("*") for a generic entry.

ASSOCIATED USER

is the RACF user identification associated with the procedure. An equal sign ("=") indicates that the procedure name is used for the RACF user identification.

ASSOCIATED GROUP

specifies the RACF group associated with the procedure. An equal sign ("=") indicates that the procedure name is used for the RACF group name.

PRIVILEGED

indicates whether the procedure has the privileged attribute. A value of YES indicates that the procedure has the attribute; a value of NO indicates it does not.

TRUSTED

indicates whether the procedure has the trusted attribute. A value of YES indicates that the procedure has the attribute; a value of NO indicates it does not.

TRACE

indicates whether the STARTED class profile has trace activated. If the attribute is activated, then when the started task is initiated, RACF issues message IRR812I to the operator to record the activity.

Report Messages

No messages appear at the end of this report.

R A C F S T A R T E D P R O C E D U R E S T A B L E R E P O R T
FROM THE STARTED PROCEDURES TABLE (ICHRIN03):

PROCEDURE NAME	ASSOCIATED USER	ASSOCIATED GROUP	PRIVILEGED	TRUSTED
JES2	STCUSER	STCGROUP	NO	YES
IRRDPTAB	STCUSER	STCGROUP	NO	NO
IEEVMPCR	STCUSER	STCGROUP	NO	YES
APSWPROC	STCUSER	STCGROUP	NO	YES
VTAM	STCUSER	STCGROUP	NO	YES
LLA	STCUSER	STCGROUP	NO	YES
LLAEP	STCUSER	STCGROUP	NO	YES
RPCD	RPCD	STCGROUP	NO	YES
SECCLNTD	SECCLNTD	STCGROUP	NO	YES
SECD	SECD	STCGROUP	NO	YES
RSFJ	STCUSER	STCGROUP	NO	NO
RSFK	STCUSER	STCGROUP	NO	NO
RSFL	STCUSER	STCGROUP	NO	NO
*	=		NO	NO

Figure 24. Sample RACF Started Procedures Table Report (ICHRIN03)

R A C F S T A R T E D P R O C E D U R E S T A B L E R E P O R T
 FROM PROFILES IN THE STARTED CLASS:

PROFILE NAME	ASSOCIATED USER	ASSOCIATED GROUP	PRIVILEGED	TRUSTED	TRACE
CICS.REGIONA	CICSA		NO	NO	NO
CICS.REGIONB	CICSB		NO	NO	NO
NOSTDATA.NOSTDATA	-STDATA NOT SPECIFIED, ICHRIN03 WILL BE USED-				
NOUSER.JOBX	-USER NOT SPECIFIED, ICHRIN03 WILL BE USED-				
ANETVIEW.* (G)	STCUSR	SYS1	NO	YES	NO
APPC.* (G)	STCUSR	SYS1	NO	YES	NO
APSWPROC.* (G)	STCUSR	SYS1	NO	YES	NO
ASCH.* (G)	STCUSR	SYS1	NO	YES	NO
ASCHINT.* (G)	STCUSR	SYS1	NO	YES	NO
BLSJPRMI.* (G)	STCUSR	SYS1	NO	YES	NO
CATALOG.* (G)	STCUSR	SYS1	NO	YES	NO
CDSADV.* (G)	CDSADV	SYS1	NO	NO	NO
CDSCLRK.* (G)	CDSCLRK	SYS1	NO	NO	NO
CDS.* (G)	CDS	SYS1	NO	NO	NO
DTSD.* (G)	DTSD	SYS1	NO	NO	NO
DTSTP.* (G)	DTSTP	SYS1	NO	NO	NO
DUMPSRV.* (G)	STCUSR	SYS1	NO	YES	NO
IEEVMPGR.* (G)	STCUSR	SYS1	NO	YES	NO
IRRDP TAB.* (G)	STCUSR	SYS1	NO	NO	NO
JES2.* (G)	STCUSR	SYS1	NO	YES	NO
LLA.* (G)	STCUSR	SYS1	NO	YES	NO
LLAEP.* (G)	STCUSR	SYS1	NO	YES	NO
NETVFCT.* (G)	STCUSR	SYS1	NO	NO	NO
NETVREL1.* (G)	STCUSR	SYS1	NO	NO	NO
NETVREL2.* (G)	STCUSR	SYS1	NO	NO	NO
NETVREL3.* (G)	STCUSR	SYS1	NO	NO	NO
NETVSSI.* (G)	STCUSR	SYS1	NO	NO	NO
NEV313.* (G)	STCUSR	SYS1	NO	NO	NO
RACF.* (G)	STCUSR	SYS1	NO	NO	NO
RPCD.* (G)	RPCD	SYS1	NO	NO	NO
RSFJ.* (G)	STCUSR	SYS1	NO	NO	NO
RSFK.* (G)	STCUSR	SYS1	NO	NO	NO
RSFL.* (G)	STCUSR	SYS1	NO	NO	NO
RUNJOB.* (G)	STCUSR	SYS1	NO	NO	NO
SECCLNTD.* (G)	SECCLNTD	SYS1	NO	NO	NO
SECD.* (G)	SECD	SYS1	NO	NO	NO
SMF.* (G)	STCUSR	SYS1	NO	YES	NO
TCAS.* (G)	STCUSR	SYS1	NO	NO	NO
TSOCMD.* (G)	STCUSR	SYS1	NO	NO	NO
TSODB.* (G)	STCUSR	SYS1	NO	NO	NO
TSOICMD.* (G)	STCUSR	SYS1	NO	NO	NO
VLF.* (G)	STCUSR	SYS1	NO	YES	NO
VTAM.* (G)	STCUSR	SYS1	NO	YES	NO
** (G)	=MEMBER	STCGRP	NO	NO	YES

Figure 25. Sample RACF Started Procedures Table Report (STARTED Class Active)

Selected user attribute report

The selected user attribute report lists all RACF users with the SPECIAL, OPERATIONS, AUDITOR, or REVOKE attribute and indicates whether a user possesses the attribute on a system (user) or group level.

You can use the selected user attribute report to verify that only those users who need to be authorized to perform certain functions have been assigned the corresponding attribute.

Column Headings

USERID

is the user's system identifier.

ATTRIBUTE TYPE

identifies each attribute and indicates whether the user has the attribute on a system (user) or a group level. SYSTEM indicates the user has that attribute on a system level, or at all times. GROUP indicates user has the attribute only within one or more of the groups to which the user is connected. If neither SYSTEM nor GROUP appears, the user does not possess that attribute on either level.

If a user has one or more attributes on a group level, you can determine the names of the corresponding group or groups through the LISTUSER command or the "User Services" panel.

The report lists the following attribute types:

SPECIAL

gives the user complete control over all the RACF profiles in the RACF database and authority to issue all RACF commands, except those reserved for the auditor's use.

OPERATIONS

gives the user authority to perform maintenance operations and provides full authority to access RACF-protected DASD data sets and certain resource classes.

AUDITOR

gives the user complete authority to audit security controls and the use of system resources.

REVOKE

prevents, on a system level, a RACF-defined user from entering the system at all. On a group level, a user can enter the system but cannot use any group authorities associated with the group, or access data sets using that group's authority.

Note: When REVOKE is specified with a future date, the status change does not occur until the specified date. Until that date, the report does not list the user as revoked.

For more information on each attribute, especially at the group level, see *z/OS Security Server RACF Security Administrator's Guide*.

ASSOCIATIONS

are the characteristics of the user ID association. The report lists the following associations:

NODE.USERID

is the node (local or remote) and user ID of the associated user

PASSWORD SYNC

tells whether password synchronization has been requested between the listed user and associate user

ASSOCIATION TYPE

describes the type of association, the status of the user ID association, or both

Report Messages

The following message may appear below the report column headings:

NO SELECTED USERS FOUND

Explanation: There are no users with the SPECIAL, OPERATIONS, AUDITOR, or REVOKE attributes on either a system or group level.

Note: Under normal circumstances, this message should not appear. At least one user should have the SPECIAL attribute on a system level, and at least one user should have the AUDITOR attribute on a system level. If this message appears, notify your RACF security administrator or your installation manager.

```

-----
S E L E C T E D   U S E R   A T T R I B U T E   R E P O R T
USERID  -----  ATTRIBUTE TYPE -----  -----  ASSOCIATIONS -----
SPECIAL  OPERATIONS  AUDITOR  REVOKE  NODE.USERID  PASSWORD  ASSOCIATION
                                           SYNC  TYPE
-----
JPETUSR  SYSTEM      SYSTEM      SYSTEM

```

Figure 26. Selected User Attribute Report

Selected user attribute summary report

The selected user attribute summary report shows totals for installation-defined users and for users with the SPECIAL, OPERATIONS, AUDITOR, and REVOKE attribute at both the system and the group level. You can use the summary report to verify that the number of users with each of the selected attributes, on either a system or a group level, is the number your installation wants.

Note: The selected user attribute summary report is produced automatically after the selected user attribute report; it cannot be requested separately.

Column Headings

TOTAL DEFINED USERS

is the number of users defined by your installation.

TOTAL SELECTED ATTRIBUTE USERS

is the number of users with each of the four selected attributes (SPECIAL, OPERATIONS, AUDITOR, and REVOKE) at both the system and group level.

Report Messages

No messages appear at the end of this report.

```

-----
S E L E C T E D   U S E R   A T T R I B U T E   S U M M A R Y   R E P O R T
-----
TOTAL DEFINED USERS:                1
TOTAL SELECTED ATTRIBUTE USERS:
ATTRIBUTE BASIS  SPECIAL  OPERATIONS  AUDITOR  REVOKE
-----
SYSTEM          1          1          1          0
GROUP           0          0          0          0

```

Figure 27. Selected User Attribute Summary Report

Selected data sets reports

The selected data sets report lists all the data sets, including the RACF database or databases, that meet one or more of the selection criteria that DSMON uses. For each selected data set, the report specifies the serial number of the volume on which the data set resides, the selection criterion, whether the data set is RACF-indicated or RACF-protected, and the universal access authority (UACC) for the data set. If a data set or RACF database meets more than one selection criterion, there is a separate entry for each criterion.

You can use the selected data sets report to determine which system and RACF data sets are protected by RACF and which are not. You can also check to learn whether the UACC associated with each of the data sets is compatible with the resource access control requirements of your installation.

Column Headings

DATA SET NAME

is the name of the data set.

VOLUME SERIAL

is the serial number of the direct access volume on which the data set resides. If the data set is not cataloged, this column is blank.

SELECTION CRITERION

is the criterion that was used to select the data set for the report.

The following entries may appear:

LNKLST

The data set is part of the LNKLST concatenation (which is SYS1.LINKLIB and any data sets concatenated to SYS1.LINKLIB by using the LNKLSTxx member of SYS1.PARMLIB) for this IPL.

APF

specifies that the data set is an APF-authorized library.

For information about defining the format and contents of the list of APF-authorized libraries used by MVS, refer to *z/OS MVS Initialization and Tuning Reference*.

Note:

1. Depending on your APF list definition, the list of APF-authorized libraries may be incomplete in the Selected Data Sets report generated by the FUNCTION ALL or FUNCTION SYSAPF control statements. Only APF-authorized libraries contained in the IEAAPFxx or PROGxx members of SYS1.PARMLIB, or specified by the MVS SETPROG operator command, are reflected in the report. APF-specification members can come from any member of PARMLIB, or from a command which dynamically adds APF data sets. Therefore, LPA, MLPA, and FLPA authorized libraries that are not defined in the APF LIST are not flagged as APF.
2. Perform one of the following options to include all APF-authorized libraries in the Selected Data Sets report.
 - Define all your LPA, MLPA, and FLPA libraries in the applicable IEAAPFxx or PROGxx members of SYS1.PARMLIB. This allows MVS to recognize them as APF-authorized at all times.

- Use the FUNCTION USRDSN and USEROPT USRDSN control statements and specify the APF-authorized libraries that are not defined in the APF List. With this option, USRDSN is the SELECTION CRITERION field,
- If the APF-authorized library is part of the LNKLIST concatenation, specify either FUNCTION ALL or FUNCTION SYSLNK. In this case, the SELECTION CRITERION field contains LNKLIST-APF.

LNKLST-APF

specifies that the data set is a linklist data set that is also an APF authorized library.

MASTER CATALOG

indicates that the data set is the MVS master catalog.

USER CATALOG

indicates that the data set is a user catalog.

RACF PRIMARY

indicates that the data set is a primary RACF database, containing RACF access control information. This information includes user, group, data set, and general-resource profiles.

RACF BACKUP

indicates that the data set is a backup or recovery RACF database.

SYSTEM

indicates that the data set is one of the following system data sets:

- SYS1.CMDLIB
- SYS1.LINKLIB
- SYS1.LPALIB
- SYS1.NUCLEUS
- SYS1.PARMLIB
- SYS1.PROCLIB
- SYS1.SVCLIB
- SYS1.UADS

USRDSN

is the user data set specified on the USEROPT control statement.

RACF INDICATED

indicates whether the data set is RACF-indicated.

The following entries may appear:

YES

indicates that the RACF indicator for the data set is on.

NO

indicates that the RACF indicator for the data set is off. RACF will not check for a discrete profile.

N.C.

indicates that the data set is not listed (cataloged) in the master catalog.

N.M.

indicates that the DASD volume on which the data set resides is not mounted or has been dynamically deleted.

N.F.

indicates DSMON cannot find the data set on the specified volume. For APF data sets, this may indicate a security exposure that should be investigated and corrected.

RACF PROTECTED

indicates whether the data set has a RACF profile. The following entries may appear:

YES

indicates that the data set has a discrete or generic profile. If the RACF indicator for the data set is off, the data set is protected by a generic profile.

NO indicates that no profile exists for the data set. The data set is not protected in any way by RACF.

Note:

1. An error condition exists when the RACF indicator for a data set is on but no profile exists for the data set. The data set is not accessible until the condition is corrected.
2. For a data set profile that has WARNING set, RACF issues a warning message, but permits access to the data set. Thus, although the data set has a RACF profile and is indicated as RACF-protected in the report (YES), it can nevertheless be accessed and is not really protected. You may want to list the contents of the data set profile (through the LISTDSD command) to see whether WARNING is set.

UACC

is the data set's universal access authority (UACC), if it is defined. The UACC is the default access authority that specifies how the data set can be accessed by users or groups not in the access list of the data set's RACF profile.

Note: The UACC does not necessarily indicate the actual authority that a user has to access the data set. The global access checking table may contain an entry applicable to the data set, or the user may be on the access list, if the data set has a discrete profile.

The following universal access authorities may appear:

ALTER

For a data set that is protected by a discrete profile, ALTER allows all users to read, update, or delete the data set.

CONTROL

For VSAM (virtual storage access method) data sets, CONTROL provides all users with the same authority that is provided with the VSAM CONTROL password; that is, authority to perform control interval access (access to individual VSAM data blocks), and to retrieve, update, insert, or delete records in the specified data set.

For non-VSAM data sets, CONTROL is equivalent to UPDATE.

UPDATE

allows all users to read or update the data set. UPDATE does not, however, authorize a user to delete the data set.

READ

allows all users to access the data set for reading or copying only.

NONE

does not allow users to access the data set.

Report Messages

The following message may appear below the report column headings:

NO SELECTED DATA SETS FOUND

Explanation: DSMON did not find any data sets meeting the criteria.

Note: Under normal circumstances, this message should not appear. If it does, notify your RACF security administrator or installation manager.

DATA SET NAME	SELECTED	DATA	SETS	REPORT		
	VOLUME SERIAL	SELECTION CRITERION	RACF INDICATED	RACF PROTECTED	UACC	
CATALOG.AOCSMP.USERCAT	D94HV2	USER CATALOG	N.M.	YES	UPDATE	
CATALOG.CICSCAT	D83CL2	USER CATALOG	NO	YES	NONE	
CATALOG.CICSDCT	D83HW1	USER CATALOG	NO	YES	NONE	
CATALOG.CMMA00	CMMA00	USER CATALOG	N.M.	YES	NONE	
CICS.CURRENT.SDFHAUTH	DB2LD1	APF	NO	YES	NONE	
CICS.CURRENT.SDFHEXC1	DB2LD1	APF	NO	YES	READ	
CICS.CURRENT.SEYUAUTH	DB2LD1	APF	NO	YES	NONE	
CICS.NEW.SDFHAUTH	DB2LD1	APF	NO	YES	NONE	
CICSCAT.USERCAT	CIC330	USER CATALOG	N.M.	NO		
DB11.SDSNEXT	SMS066	APF	NO	YES	READ	
DB2.DB2L.DB2PM.SDG0LOAD	DB2LS2	APF	NO	YES	READ	
DB2.DB2M.DB2PM.SDG0LOAD	DB2S01	APF	NO	YES	READ	
DB2.LOCAL.LOADLIB	DB2LS0	APF	NO	YES	READ	
DB2.USER.CATALOG	DB2PR0	USER CATALOG	NO	YES	NONE	
DB2L.DSNEXIT	DB2LS1	APF	NO	YES	READ	
DB2M.DSNEXIT	DB2M50	APF	NO	YES	READ	
LINKLIST.DB2L.DSNLINK	DB2LS1	APF	NO	YES	READ	
		LNKLIST - APF				
LINKLIST.DB2L.DSNLOAD	DB2LS1	APF	NO	YES	READ	
		LNKLIST - APF				
LINKLIST.DB2M.DSNLINK	DB2M50	APF	NO	YES	READ	
		LNKLIST - APF				
LINKLIST.DB2M.DSNLOAD	DB2M50	APF	NO	YES	READ	
		LNKLIST - APF				
LINKLIST.DB2PM.SDG0LINK	DB2M50	APF	NO	YES	READ	
NCPI.SSPLIB	TPPAK2	APF	NO	YES	READ	
NETVIEW.NETVIEW.USERLINK	TPPAK5	APF	NO	YES	NONE	
NETWORK.ANOMVS.USERLINK	TPPAK5	APF	NO	YES	READ	
POSIX.CBC.OSVIR2M0.USERCAT	BPXRTL	USER CATALOG	NO	YES	NONE	
SYS1.ACCOUNT	PPP002	APF	YES	YES	NONE	
		LNKLIST - APF				
SYS1.AUTHLIB	PPP002	APF	YES	YES	READ	
		LNKLIST - APF				
SYS1.CEE.SCEEKED	PR1PK3	APF	NO	YES	READ	
SYS1.CEE.SCEERUN	PR1PK3	APF	NO	YES	READ	
		LNKLIST - APF				
SYS1.CICS410.LINKLIB	DB2LD1	APF	NO	YES	NONE	
		LNKLIST - APF				
SYS1.CICS410.LPALIB	DB2LD1	APF	NO	YES	NONE	
SYS1.CMDLIB	PR1PK3	APF	NO	YES	READ	
		LNKLIST - APF				
		SYSTEM				

Figure 28. Sample Selected Data Sets Report

Appendix A. The RACF report writer

Attention:

The report writer is *no longer* the recommended utility for processing RACF audit records. The RACF SMF data unload utility is the preferred reporting utility. The report writer does not support all of the audit records introduced after RACF 1.9.2. See Chapter 3, “The RACF SMF data unload utility,” on page 49, for more details.

The RACF report writer (RACFRW) uses SMF dates in the form *yyddd*. If you attempt to select a date range of records with a starting date that occurs before January 1, 2000 (for example, 99364) and the ending date occurs on or after January 1, 2000 (for example, 00002) the report writer will reject your request as it will consider the year 00 as coming before the year 99. Similarly, when sorting records by date, the report writer will treat 00 as coming before 99. IBM does not intend to enhance the RACF report writer to recognize this condition and to process the records differently, as IBM has stabilized RACFRW and will not make functional improvements to it. Other than this problem with record ordering, which should only occur if the input file has records both before and after January 1, 2000, RACFRW should properly process records with dates after January 1, 2000, if it would have handled those records if they had contained earlier dates.

A successful security mechanism requires that appropriate personnel, particularly the auditor and the security administrator, be able to assess the implementation of the security mechanism and the use of the resources it protects. The RACF report writer provides a wide range of reports that enable you to monitor and verify the use of the system and resources.

The RACF report writer lists the contents of system management facilities (SMF) records in a format that is easy to read. SMF records reside in the SMF data file. You can also tailor the reports to select specific SMF records that contain certain kinds of RACF information. With the RACF report writer, you can obtain:

- Reports that describe attempts to access a particular RACF-protected resource in terms of user name, user identity, number and type of successful accesses, and number and type of attempted security violations.
- Reports that describe user and group activity.
- Reports that summarize system use and resource use.

How the RACF report writer operates

The RACF report writer consists of three phases:

- Command and subcommand processing
- Record selection
- Report generation

See Figure 29 on page 120 for an overview of the RACF report writer. The figure also shows the replaceable module, ICHRSMFI, for the RACF report writer, and the RACF report writer installation-wide exit, ICHRSMFE.

ICHRSMFI is a nonexecutable module that contains default values for the RACF report writer sort parameters, dynamic-allocation parameters, and processing options. See *z/OS Security Server RACF System Programmer's Guide* for a description of the contents of the module and an explanation of how to modify the module if necessary.

ICHRSMFE is an installation-wide exit that the RACF report writer calls during the record selection phase. The exit allows you to add functions such as the following to the RACF report writer:

- Create additional selection and or rejection criteria (or both) for records that the RACF report writer processes
- Modify naming conventions in records that the RACF report writer processes
- Add other reports to those that the RACF report writer provides.

Detailed information about coding the ICHRSMFE exit routine appears in *z/OS Security Server RACF System Programmer's Guide*.

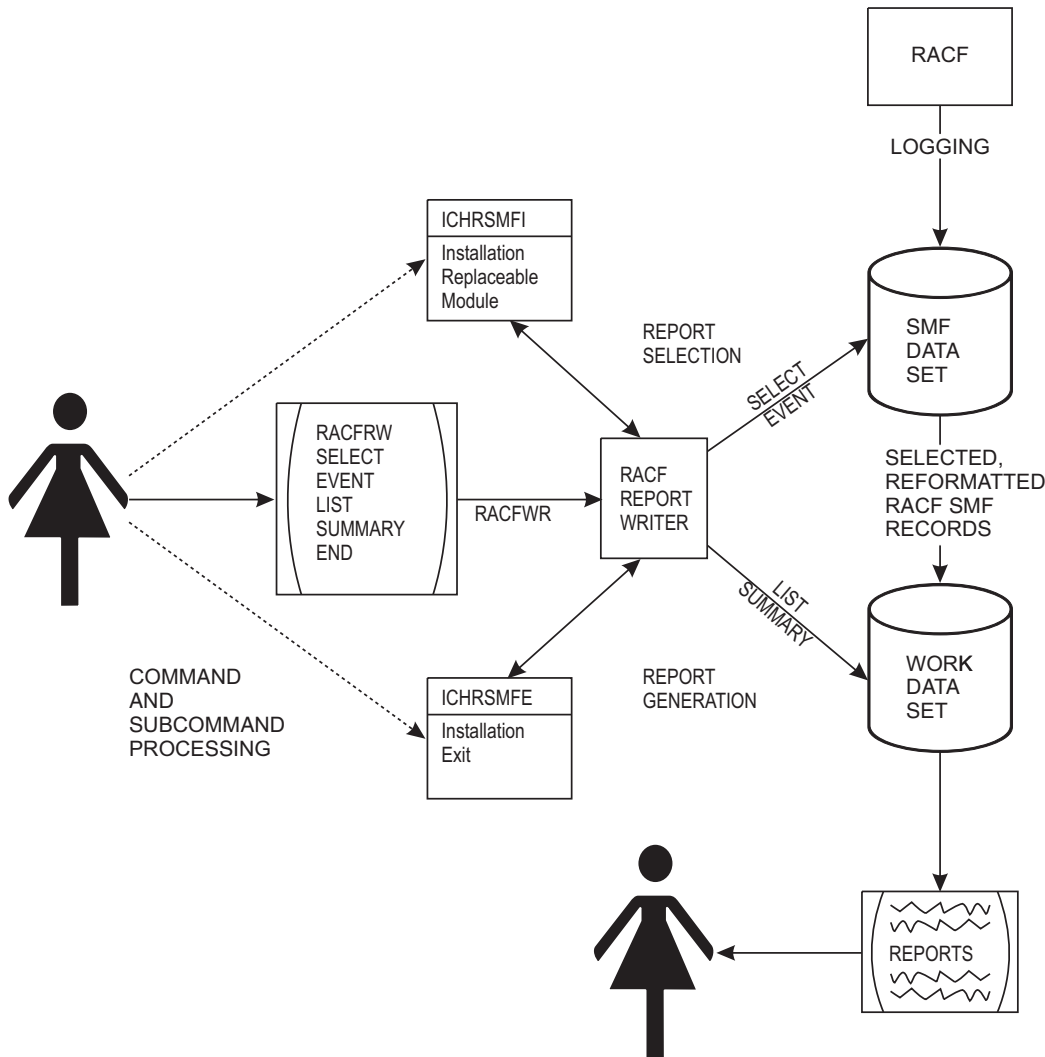


Figure 29. RACF Report Writer Overview

Phase 1

Command and subcommand processing

The first phase, *command and subcommand processing*, starts when you enter the TSO command RACFRW or run the report writer as a batch job. As a command, RACFRW invokes the RACF report writer through the terminal monitor program (TMP) and places you in subcommand mode. In subcommand mode, you can

enter the RACF report writer subcommands SELECT, EVENT, LIST, SUMMARY, and END. When the RACF report writer is invoked from a batch job, the batch job invokes the TMP through a job step in the JCL, and RACFRW commands and subcommands can be specified as data in stream to the job. See “The RACF report writer and the SMF input data set” on page 124.

Briefly, the SELECT and EVENT subcommands specify which of the input records the RACF report writer selects and uses to generate the reports. You can then produce those reports by using the LIST subcommand to format and print a listing of each SMF record you select and the SUMMARY subcommand to format and print a summary listing of the SMF records. After entering all the subcommands you need, enter the END subcommand. END terminates subcommand mode and the first processing phase.

Note: Pressing PA1 or the attention key at any time during this first phase terminates the RACF report writer immediately and returns control to the TMP.

Phase 2

Record selection

During the second phase, *record selection*, the RACF report writer compares each record from the input file—the SMF records—against the criteria you specify on the SELECT and EVENT subcommands. The RACF report writer accepts as input only RACF-related SMF records. These are process records (SMF type 20, 30, 80, and 83) and status records (SMF type 81). In addition, the report writer generates a “fake” type 81 record for every SMF type 80 record that results from a SETROPTS or RVARY command.

For a description of SMF record types 20 and 30, see *z/OS MVS System Management Facilities (SMF)*. For a description of SMF record types 80, 81, and 83, see *z/OS Security Server RACF Macros and Interfaces*.

Note:

1. The SMF type 81 record contains “UCB” instead of an EBCDIC device name if the master RACF primary database is on a device with an address greater than X'FFF'. When the RACF report writer formats the type 81 record, this information is displayed for you to see.
2. The SMF type 83 subtype 1 record is generated when SETROPTS MLACTIVE is in effect and a RACF command (ALTDSD, ADDSD, DELDSD) has changed the security label in a profile. The record contains the names of the cataloged data sets affected by the security-label change. A link value is contained in both the SMF type 80 record for the RACF command and the SMF type 83 subtype 1 record. The link value is used to connect the list of data set names affected by the security-label change with the RACF command that caused the change. The text in the report-writer output is “LINK=*numeric value*”.

If there are migrated items in the list, and the migration facility is unavailable at the time the command is issued, the following messages will be printed after the items:

```
** Unable to verify this  
** migrated item.(1)
```

The number in parentheses denotes diagnostic information used by IBM support.

For more information about using the LISTDSD command, see *z/OS Security Server RACF Command Language Reference*.

If you do not specify any SELECT or EVENT subcommands, the RACF report writer selects all of the records from the input file for further processing. If you specify options that limit your report, only limited information is saved.

Record reformatting

To sort and print the SMF input records, the RACF report writer must reformat them. The report writer allocates an in-storage buffer for reformatting, using it on each SMF record being processed. The size of this buffer is determined by the WRKRECL field in the installation-replaceable module ICHRSMFI unless LRECL is specified on SORTIN DD. The LRECL value in the SORTIN DD statement overrides the WRKRECL statement used by RACFRW.

The report writer makes sure that the buffer is large enough for the base section of the SMF record. However, it does not guarantee that the relocate sections of the SMF record will fit.

In the report writer output, the process records that do not fit into the buffer are noted as *truncated*. The status records that do not fit will be noted as *bypassed*. The WRKRECL default is 4096.

The RACF report writer copies the reformatted records to a work data set. You can save this work data set and use the reformatted records as input to a later run of the RACF report writer.

If the input consists of records previously saved using the report writer, those records are already reformatted. The RACF report writer skips the reformatting step for those records. Operands on the RACFRW command specify whether the RACF report writer is to reformat the input records and whether the work data set is to be saved for subsequent runs of the RACF report writer.

When the RACF report writer has compared all the input records against the selection criteria and, if necessary, has reformatted the selected records and copied them to a work data set the second processing phase is complete.

Phase 3

Report generation

During the third phase, *report generation*, the RACF report writer generates the reports that you request with the LIST and SUMMARY subcommands. It uses as input only the records from the work data set. The RACF report writer always produces a header page with a list of the subcommands that you have entered and describes the meanings of values for such activities as job initiation, TSO logon, resource access, and use of RACF commands that appear in the reports. The other reports depend on operands you have specified, but the RACF report writer always produces the reports you request according to a specific order. See the examples at the end of this chapter .

If you want a general summary report of overall system activity related to RACF, you can specify the GENSUM operand on the RACFRW command. The RACF report writer:

1. Collects the data for the general summary report during the record selection phase (see “Phase 2” on page 121) and prints it before any other reports during phase 3.
2. Produces reports for the LIST subcommand and lists all SMF records from the work data set in the sequence that you specified.
3. Produces a separate summary report of the SMF records for each SUMMARY subcommand you enter with a RACFRW command. Depending on the subcommand you enter, the report contains records by group, resource, command, RACF event, or owner activity.

Sample reports produced by GENSUM, LIST, and SUMMARY are shown in the section “Sample reports” on page 152. When it has completed the last report, the RACF report writer terminates and returns control to the TMP.

RACF report writer command and subcommands

The following tables summarize the main RACFRW command operands and subcommands that control report writer processing:

Table 12. Summary of RACFRW Command and Its Operands

Operand	Result
GENSUM	Produces a general summary report of system activity related to RACF
NOGENSUM	Produces no general summary report
FORMAT	Specifies that SMF records are to be formatted for use by the report writer
NOFORMAT	Specifies that the input SMF records are already formatted for use by the report writer; no reformatting is necessary
SAVE	Saves the reformatted records on a work data set. Only those records that satisfy the specified SELECT/EVENT criteria are saved

Table 13. Summary of RACFRW Subcommands

Subcommand	Result
SELECT	Specifies which SMF records to choose from the input file for report writer processing
EVENT	Specifies further which SMF records to choose from the input file; for the report writer to process these records, each record must meet the criteria
LIST	Specifies that the report writer is to list each record that is processed by SELECT/EVENT groups
SUMMARY	Specifies that the report writer is to print summary reports for records processed by SELECT/EVENT groups
END	Terminates subcommand processing

Planning considerations

To use the RACF report writer at your installation, you must have:

- The DFSORT IBM Program Product (Program Number 5740-SM1), or equivalent.
- An output device that can handle 133 character lines.

The RACF report writer and the SMF input data set

The input data set to the RACF report writer consists of the following SMF record types:

- 20 Job initiation
- 30 Common address work data
- 80 RACF processing
- 81 RACF initialization
- 83 RACF processing

Attention:

Even though some commands use the relocate 44 section of the record, the output of these records is not consistent. The RACF SMF data unload utility is the preferred reporting utility.

SMF records

Records from the SMF data set or log stream must first be dumped to a data set that RACF can use as input. If you have access to the SMF data set or log stream, you can use the SMF dump program (IFASMFDP or IFASMF DL) to dump the SMF records. (If your installation does not allow you to access the SMF data set or log stream, see your SMF system programmer to find out how you can obtain the SMF records as input to the RACF report writer.)

Running the report writer as a batch job

For large SMF data sets, you should run the report writer as part of a batch job. The following JCL is an example of how to dump the SMF records to a temporary data set and run the report writer as a batch job.

In Figure 30 on page 125, the SMF dump program IFASMFDP dumps record types 20, 30, 80, 81, and 83 from an SMF data set (SYS1.MANA) to a temporary data set (QSAMOUT DD) for use by the report writer.

```

/*****
/*****
/*
/*          RUN THE SMF DUMP PROGRAM.
/*
/*
/*****
/*****
//SMFDUMP EXEC PGM=IFASMFDP
//SYSPRINT DD SYSOUT=*
//VSAMIN  DD DSN=SYS1.MANA,DISP=SHR
//QSAMOUT DD DSN=##QSAMOUT,DISP=(NEW,PASS,DELETE),
//        SPACE=(TRK,(25,50),RLSE),UNIT=SYSALLDA
//SYSIN   DD *
           INDD(VSAMIN,OPTIONS(DUMP))
           OUTDD(QSAMOUT,TYPE(020,030,080,081,083))
           DATE(89195,89195)
           SID(MVS1)
           SID(MVS3)
/*****
/*****
/*
/*          RUN THE RACF REPORT WRITER AS A BATCH JOB
/*          AND USE SMF DATA FROM QSAMOUT.
/*
/*
/*****
/*****
//RACFRW2 EXEC PGM=IKJEFT01
//SORTWKxx DD your sort work files
//SYSPRINT DD SYSOUT=*
//SYSTSPRT DD SYSOUT=*
//RSMFIN  DD DISP=(SHR,PASS,DELETE),DSN=*.SMFDUMP.QSAMOUT
//SYSTSIN DD *,DLM=XX
           RACFRW TITLE('RACF REPORTS') GENSUM
           SELECT VIOLATIONS
           LIST TITLE('ACCESS VIOLATIONS SUMMARY REPORT')
           SUMMARY RESOURCE BY(USER)
           END
XX

```

Figure 30. JCL for Dumping SMF Records and Running the Report Writer as a Batch Job

You can specify options for IFASMFDP on the SYSIN INDD statement, and the selection criteria for the SMF records on the SYSIN OUTDD statement. You can also specify the start and end date for the dump program in Julian format (YYDDD) on SYSIN DATE and the system identification on SYSIN SID.

For more information about IFASMFDP and the SMF dump options, including outputting log stream output using IFASMF DL, see *z/OS MVS System Management Facilities (SMF)*.

RACFRW then uses the temporary data set QSAMOUT as input defined on the RSMFIN DD statement, and you can specify the report-writer command and subcommands as in-stream data to SYSTSIN DD.

Running the report writer using the RACFRW command

You can also run the RACF report writer as a TSO command. In TSO ready mode enter **RACFRW**. RACF places you in subcommand mode, and you can enter the report writer subcommands (SELECT, EVENT, LIST, SUMMARY, and END).

If you run the report writer as a TSO command, you must pre-allocate the data set that contains the selected SMF records as RSMFIN and use it as input to the report writer command and subcommands. See “Pre-allocating data sets” on page 126 for

more information about pre-allocating data sets for the report writer.

Pre-allocating data sets

If you run the report writer as a TSO command, pre-allocate the data sets required by the RACF report writer using the following ddnames:

RSMFIN

The input data set or sets. Note, however, that if you enter the DATASET operand on the RACFRW command, the RACF report writer assigns a system-generated DD name to this input data set and ignores RSMFIN. If you neither pre-allocate the input data set nor specify the DATASET operand, the RACF report writer issues message ICH64305I, and terminates immediately.

SYSPRINT

The output data set. If you do not pre-allocate this output data set, the RACF report writer allocates this data set to a SYSOUT data set (which goes to the terminal on which you are entering the commands and subcommands).

SORTIN

The work data set. If you enter the SAVE operand on the RACFRW command, the RACF report writer assigns SORTIN to the work data set that you specify in the SAVE operand. If you pre-allocate the work data set or specify the SAVE operand, the RACF report writer saves this work data set for future use; otherwise, it allocates the work data set to a temporary data set and deletes it at job termination. See the SAVE and FORMAT/NOFORMAT options described in "RACFRW command" on page 128.

If the logical record length is specified, it overrides the WRKLRECL field in the installation-replaceable ICHRSMFI module. The default value of WRKLRECL is 4096. If the logical record length you specify is not large enough to hold the largest SMF record from RSMFIN, the report writer truncates the record, losing some of the information for the record's output.

SORTLIB

The system library that contains the SORT/MERGE load modules. If you do not pre-allocate this system library, the RACF report writer allocates it to the sort data set named in SORTDSN in ICHRSMFI. Initially, the name in SORTDSN is SYS1.SORTLIB.

SORTDDNM

The SORT/MERGE messages. The RACF report writer allocates these messages to the data set named in SORTDDNM in ICHRSMFI. If you do not pre-allocate these messages, they go to the terminal on which you are entering the commands and subcommands, because the initial name in SORTDDNM is SYSOUT.

SORTWK_{xx}

The SORT/MERGE work file(s), named SORTWK01 to SORTWK_{nn}. If you do not pre-allocate these files, dynamic allocation occurs, using the dynamic allocation parameter specified in SORTDYN in ICHRSMFI. Initially, SORTDYN contains 'DYNALLOC=3350'.

Note that any data set that you pre-allocate remains allocated after the RACF report writer terminates, while any data set allocated by the RACF report writer is deallocated before termination.

RACF report writer return codes

After completing, the RACF report writer returns control to the terminal monitor program (TMP) with a return code in register 15.

The following are possible return codes:

Return Code

Meaning

- | | |
|----|---|
| 0 | The report writer has terminated normally. |
| 12 | The report writer has not terminated successfully for one of the following reasons: <ul style="list-style-type: none">• It could not dynamically allocate any needed resource that was not pre-allocated by the user• It could not open any needed resource• It received a nonzero return code from a service routine that it has invoked• It received a nonzero return code from the SORT/MERGE routines. |

If you receive a return code of 12, check to see whether any error messages were issued when you invoked the report writer.

- If you receive a return code of 12 when the report writer is running in batch, check that the job statement in the JCL specifies MSGLEVEL=(1,1).
- If you receive a return code of 12 when you invoke the report writer from a TSO terminal, make sure the following option is included in your user profile:

```
profile wtpmsg msgid
```

For more information about report writer error messages, see *z/OS Security Server RACF Messages and Codes*.

Useful hints

When you use the RACF report writer, consider the following:

- You must use the SMF dump program, IFASMFDP, to dump the SMF data set, which is a VSAM data set, into a QSAM data set, which is what the RACF report writer requires. For additional information about IFASMFDP, see *z/OS MVS System Management Facilities (SMF)*.
- In an installation using RACF to protect multiple systems, each system writes RACF-generated SMF records to a different data set. You can concatenate all of these data sets into a single data set for input to the RACF report writer. Later, should you have to separate the information based on the identifier of the system that generated it, you could use the SYSID operand on either the LIST or the SELECT subcommand.
- By using the SELECT and EVENT subcommands, you can retrieve individual SMF records of interest for display at a TSO terminal (display screen).
- If your SMF file is large or resides on multiple tape volumes, you may consider specifying the SAVE operand for the work data set that you create. This action reduces the amount of time and number of devices you need, should you need to use this work data set again to produce additional reports. Note that by using SELECT and EVENT subcommands, you can create and save a subset of a work data set that you saved in a previous run of the RACF report writer.

- Your system programmer can provide special SMF record selection and tailoring by using the RACF report-writer exit routine ICHRSMFE. For more information, see *z/OS Security Server RACF System Programmer's Guide*.
- The RACF report writer runs as a postprocessor of RACF and does not interfere with normal RACF processing.

RACFRW command

This section shows the function and syntax of the RACF report writer command (RACFRW) and subcommands (SELECT, EVENT, LIST, SUMMARY, and END). The command and subcommands are not listed alphabetically, but in the order in which you are likely to enter them. This order is: RACFRW, SELECT, EVENT, LIST, SUMMARY, and END.

The following key defines the symbols used to represent the syntax of the command and subcommands:

UPPERCASE

characters must appear as shown

lowercase

characters indicate that the user supplies the information

list... indicates that the item can be listed more than once

{ } group alternative items; you can only specify one item

[] indicates an optional item that you can specify

KEYWORD

indicates the default when no item is specified

Figure 31. Key to Symbols in Command Definitions

The TSO command RACFRW invokes the RACF report writer. After you enter the RACFRW command, TSO places you in subcommand mode and prompts you to enter the RACF report-writer subcommands until you enter the END subcommand.

On the RACFRW command, you can specify the source and disposition of input records, the data to be passed to the installation-wide exit routine (ICHRSMFE), whether the RACF report writer is to reformat the input records, and whether the RACF report writer is to print a general summary report. (See *z/OS Security Server RACF System Programmer's Guide* for further information about the installation-wide exit ICHRSMFE.)

The Syntax of the RACFRW Command:

```
RACFRW      [TITLE('q-string')]
            [DATA('q-string')]
            [{FORMAT }]
            [{NOFORMAT}]
            [{DSNAME }] (name-list...)
            [{DATASET}]
            [SAVE(name)]
            [LINECNT( { 60 } ) ]
```


[{number}]

[{GENSUM }]
[{}NOGENSUM}]

TITLE('q-string')

specifies a string of up to 132 characters, enclosed in single quotation marks, to be used as a default heading for the report pages, if the TITLE operand on either the SUMMARY or LIST subcommand does not specify a unique report heading for a requested report.

DATA('q-string')

specifies a string of up to 256 characters of data, enclosed in single quotation marks, to be passed to the installation-wide exit routine (ICHRSMFE).

FORMAT

specifies that the RACF SMF records used as input to the RACF report writer must be reformatted (from the way they appear in the SMF records) before processing. For additional information about the reformatted records, see *z/OS Security Server RACF System Programmer's Guide*. FORMAT implies that the RACF report writer has not previously processed the input records. FORMAT is the default value.

NOFORMAT

specifies that the RACF SMF records used as input to the RACF report writer are already reformatted and suitable for processing. NOFORMAT implies that the input records have been processed previously by the RACF report writer and saved. You can save input records by specifying the SAVE operand.

Note: Specifying FORMAT for a data set that is already reformatted or specifying NOFORMAT for a data set that is not already reformatted can cause unpredictable results.

If report-writer input is from SMF, records are not reformatted. If input is a file saved from a previous report-writer run, records are reformatted.

Restriction:

If records have been reformatted and saved using the SAVE operand on one release of RACF report writer, the same release must be used to process the saved reformatted records. For example, RACF 1.9 reformatted records must be processed with RACF 1.9. SMF records from previous RACF releases, however, are supported. If you want to process SMF data from previous releases, archive the original SMF records rather than the reformatted records.

DSNAME(name-list...) or DATASET(name-list...)

specifies the name of one or more cataloged data sets to be concatenated and used as input to the RACF report writer. If you omit this operand, the RACF report writer uses as input the data set you have pre-allocated to the RSMFIN DD name. For more information about preallocating RSMFIN, see "Pre-allocating data sets" on page 126.

SAVE(name)

specifies the name of a sequential data set to be assigned to the work data set that is to contain the selected, reformatted RACF SMF records. If this 'name' data set is new, the RACF report writer allocates and catalogs it. If this 'name' data set is old, the RACF report writer replaces the data currently in the data set with the new data and keeps the data set. You can use this saved work-data set as input to a later run of the RACF report writer.

If you omit this operand and have not pre-allocated a SORTIN DD name, the work-data set is deleted at job termination.

LINECNT (number)

specifies the maximum number of lines to be written before ejecting to a new page. The minimum number that you can specify is 20. If you specify a number lower than 20, LINECNT defaults to 20. If you omit this operand, LINECNT defaults to 60.

GENSUM

specifies that a general summary report is to be printed. This report contains various statistics about all the RACF SMF records processed, such as total JOB/LOGON attempts, successes, and violations, total resource accesses, successes, and violations, and a breakdown of JOB/LOGON and resource access violations by hour.

NOGENSUM

specifies that a general summary report is not to be printed. NOGENSUM is the default value.

RACFRW subcommands

When you invoke RACFRW as a TSO command, you are placed in subcommand mode. You can then enter subcommands to select the records and the format for the reports.

SELECT subcommand

The SELECT subcommand allows you to choose specific records from the input file containing the RACF SMF records. The RACF report writer reformats these selected records, if necessary, and copies them to an MVS work-data set. Although all input records are used for the general summary report, the RACF report writer can list and generate summary reports for only the records that are indicated on the SELECT subcommand. The SELECT subcommand determines which records get processed.

Note: RACF reports are only as good as the SMF records used as input to them. You need to carefully consider your installation's needs when selecting audit options and be sure the report writer has enough data to make useful reports.

SELECT/EVENT groups

SELECT and EVENT subcommands provide a way to tailor RACF report-writer output. It is easier for you to review a few, selected reports than to examine all the data at once. SELECT and EVENT commands work together to restrict the SMF records that the report writer uses for input. You can run the report writer several times on the same SMF data using different SELECT and EVENT criteria to obtain several reports on specific topics. You can issue SELECT subcommand separately or with EVENT subcommands to form what is called a SELECT/EVENT group.

For each run of the report writer, you can specify zero or more SELECT/EVENT groups. Each group consists of a SELECT subcommand followed by zero or more EVENT subcommands. A second SELECT subcommand indicates the beginning of another group.

For an SMF record to be used in a RACF report, it must meet the criteria of at least one of the SELECT/EVENT groups. The SMF record must meet all the criteria of the SELECT subcommand plus all the criteria of at least one of the EVENT subcommands in that group.

A SELECT/EVENT group must begin with a SELECT subcommand, even if it is a SELECT subcommand with no operands. You can then follow this subcommand with up to 49 EVENT subcommands that specify additional selection criteria for that group. If you do not specify an EVENT subcommand, RACF uses only the criteria from the SELECT subcommand. See "EVENT subcommand" on page 136 for more information.

If you specify multiple SELECT subcommands or SELECT/EVENT groups or both, you can specify the groups in any order. The listing and summary reports that you request, however, will reflect *all* the records that have been selected by *all* the groups, not just the records selected by one particular SELECT/EVENT group. If you do not issue any SELECT subcommands or SELECT/EVENT groups, *all* the RACF SMF records from the input file are selected.

The RACF report writer can process a maximum of 50 SELECT and EVENT subcommands. If you enter more than 50, TSO accepts only the first 50, then prompts you to enter a subcommand other than SELECT or EVENT.

The following example produces a listing of all unsuccessful logons and all successful SETROPTS commands.

```
RACFRW
SELECT VIOLATIONS
EVENT LOGON
SELECT SUCCESSES
EVENT SETROPTS
LIST
END
```

The next example provides a listing of every unsuccessful RACF event (logons, accesses, SVCs, commands) plus successful logons and successful SETROPTS commands.

```
RACFRW
SELECT VIOLATIONS
SELECT SUCCESSES
EVENT LOGON
EVENT SETROPTS
LIST
END
```

The following example results in a listing of every RACF-related SMF record.

```
RACFRW
LIST
END
```

Note: Use a comma to separate items in a list of operands for SELECT or EVENT. If you must continue items in a list on another line, use the standard TSO continuation, as in the following example:

```
SELECT DATE(89195:89197) TIME(010000:120000) USER(user1,user2,+
user3,user4,user5)
```

See the syntax of the SELECT and EVENT subcommands for those operands that allow you to specify lists of items.

The syntax of the SELECT subcommand:

```
{SELECT} [DATE {(begin-number:end-number)} ]
{SEL } [ {(number-list...) } ]

[TIME {(begin-number:end-number)} ]
[ {(number-list...) } ]

[{{VIOLATIONS}}]
[{{SUCSESSES }}]
[{{WARNINGS }}]

[{{USER(name-list...)}}]
[{{NOUSER }}]

[{{JOB(name-list...)}}]
[{{NOJOB }}]

[{{OWNER(name-list...)}}]
[{{NOOWNER }}]

[GROUP(name-list...)]

[STEP(name-list...)]

[{{STATUS}}]
[{{PROCESS}}]

[SYSID(value-list...)]

[ AUTHORITY( [NORMAL] [SPECIAL] ]
[ [OPERATIONS] [AUDITOR] ]
[ [EXIT] [FAILSOFT] ]
[ [BYPASSED] [TRUSTED]) ]

[ REASON( [CLASS] [USER] [SPECIAL] ]
[ [RESOURCE] [RACINIT] ]
[ [COMMAND] [CMDVIOL] [AUDITOR] ]
[ [SECAUDIT] [VMAUDIT] ]
[ [SECLABELAUDIT] [LOGOPTIONS] ]
[ [COMPATMODE] [APPLAUDIT]) ]

[TERMINAL(name-list...)]
```

DATE(begin-number:end-number) or DATE(number-list...)

specifies a range (in ascending order) or a list of dates in the form YYDDD that are to be selected for further processing.

TIME(begin-number:end-number) or TIME(number-list...)

specifies a range (in ascending order) or a list of times in the form HHMMSS that are to be selected for further processing.

VIOLATIONS

specifies that only records identifying security violations are to be selected for further processing. This field applies to PROCESS records only.

SUCSESSES

specifies that only records identifying successful access attempts are to be selected for further processing. SUCSESSES applies to PROCESS records only.

WARNINGS

specifies that only records for which a warning message was issued are to be selected for further processing. This field applies to PROCESS records only.

If you do not specify VIOLATIONS, SUCSESSES, or WARNINGS, none of these is used as a selection criterion.

USER(name-list...)

specifies a list of user IDs that are to be selected for further processing. USER applies to PROCESS records only. If you omit both the USER and NOUSER operands, the RACF report writer selects all records containing user IDs. (See Notes 1 on page 135 and 2 on page 136.)

NOUSER

specifies that:

- Records containing user IDs are not to be selected for further processing
- Records containing undefined users are selected. You can use the list to define those user IDs if you want.

If you omit both the USER and NOUSER operands, the RACF report writer selects all records containing user IDs. If you specify both the NOUSER and NOJOB operands, the RACF report writer ignores both operands. (See Notes 1 on page 135 and 2 on page 136.)

JOB(name-list...)

specifies a list of job names that are to be selected for further processing. JOB applies to PROCESS records only. If you omit both the JOB and NOJOB operands, the RACF report writer selects all records containing job names. (See Note 1 on page 135.)

NOJOB

specifies that records that contain job names are not to be selected for further processing. If you omit both the JOB and NOJOB operands, the RACF report writer selects all records containing job names. If you specify both the NOUSER and NOJOB operands, the RACF report writer ignores both operands. (See Note 1 on page 135.)

OWNER(name-list...)

specifies a list of resource owner names that are to be selected for further processing. OWNER applies to PROCESS records only. If you omit both the OWNER and NOOWNER operands, owner is not a selection criterion.

NOOWNER

specifies that records that contain resource owner names are not to be selected for further processing. If you omit both the OWNER and NOOWNER operands, owner is not a selection criterion.

GROUP(name-list...)

specifies a list of group names that are to be selected for further processing. GROUP applies to PROCESS records only. (See Note 1 on page 135.)

STEP(name-list...)

specifies a list of step names that are to be selected for further processing. STEP applies to PROCESS records only. (See Note 1 on page 135.)

STATUS

specifies that only STATUS records are to be selected for further processing. STATUS records are RACF SMF record types 80 (generated by the SETROPTS or RVARY command) and 81.

PROCESS

specifies that only SMF record types 20, 30, 80, and 83 are to be selected for further processing.

SYSID(value-list...)

specifies a list of system identifiers that are to be selected for further processing.

AUTHORITY(type...)

specifies a list of authority types that are to be selected for further processing. AUTHORITY applies to PROCESS records only. Type can be any of the following:

SPECIAL

Selects records produced because the user had the SPECIAL or group-SPECIAL attribute

OPERATIONS

Selects records produced when access was granted because the user had the OPERATIONS or group-OPERATIONS attribute

AUDITOR

Selects records produced because the user had the AUDITOR or group-AUDITOR attribute

EXIT Selects records produced when access was granted by an installation-wide exit routine

NORMAL

Selects records produced when access was granted for a reason other than those already listed (for example, when the user had sufficient access authority)

FAILSOFT

Selects records produced when failsoft processing was in effect

BYPASSED

Selects records produced because of accesses in which RACF authority checking was bypassed because BYPASS was specified on the user ID

TRUSTED

Selects records produced when access was granted because the user had the trusted attribute.

REASON(value...)

specifies the reasons for logging the records that are to be selected for further processing. The REASON operand applies to PROCESS records only. Its value can be any of the following:

CLASS

Selects records produced because auditing of profile changes was in effect for a particular class. This record was produced because SETROPTS AUDIT was in effect.

USER Selects records produced because auditing was in effect for the specific users. This record was produced because UAUDIT was specified for the user.

SPECIAL

Selects records produced because:

- SETROPTS SAUDIT is in effect, which produces records for RACF commands requiring SPECIAL or group-SPECIAL authority.
- SETROPTS OPERAUDIT is in effect, which produces records for resource accesses requiring OPERATIONS or group-OPERATIONS authority.

If both SAUDIT and OPERAUDIT are in effect, records for both are selected. If neither one is in effect, no records are selected.

RESOURCE

Selects records produced because auditing was in effect for the specific resource or because a RACHECK installation-wide exit routine requested auditing. (See Note 3 on page 136.)

RACINIT

Selects records produced by a RACINIT request.

COMMAND

Selects records produced by commands that are always logged.

CMDVIOL

Selects records produced because auditing of command violations was in effect. This record was produced because SETROPTS CMDVIOL was in effect.

AUDITOR

Selects records produced because auditing of the specific resource was in effect. This record was produced because GLOBALAUDIT was specified in the profile. (See Note 3 on page 136.)

SECAUDIT

Selects records produced because auditing of resources according to SECLEVEL was in effect. This record was produced because SETROPTS SECLEVELAUDIT was in effect.

VMAUDIT

Selects records produced because auditing of specific z/VM events was in effect. This record has meaning only if you are sharing a database with a z/VM system.

SECLABELAUDIT

Selects records produced because auditing of resources according to security label was in effect.

LOGOPTIONS

Selects records produced because LOGOPTIONS auditing was in effect for a particular class.

COMPATMODE

Selects records produced because SETROPTS COMPATMODE was in effect.

APPLAUDIT

Selects records produced because SETROPTS APPLAUDIT was in effect.

TERMINAL(name-list...)

specifies a list of terminal IDs that are to be selected for further processing. TERMINAL applies to PROCESS records only.

Note:

1. Users who are not defined to RACF do not have a RACF user ID. Furthermore, they cannot connect to RACF. For this reason, the RACF SMF records associated with these MVS users contain the job name in place of the user ID and the step name in place of the group name.

Specifying SELECT USER(USERA) selects records for USERA including all records that have a job name in place of a user ID. If you want only records for USERA, specify:

```
SELECT USER(USERA) NOJOB
```


Similarly, specifying SELECT GROUP(GROUPA) selects records for GROUPA, including records that have a step name in place of a group name. If you want only records for GROUPA, specify:

```
SELECT GROUP(GROUPA) STEP(any-name)
```

There is no NOSTEP parameter.

2. If the user name is available in the relocate section of SMF record type 80, RACF includes it in both the PROCESS records listing and the SUMMARY reports.
3. The RACF report writer can select a record because of either RESOURCE or AUDITOR or both RESOURCE and AUDITOR.

EVENT subcommand

The EVENT subcommand allows you to specify selection criteria related to particular RACF events. For a record to be selected for further processing by the RACF report writer, it must satisfy *all* the selection criteria that you specify on this EVENT subcommand.

You can use the EVENT subcommand only with a SELECT subcommand in a SELECT/EVENT group. With the EVENT subcommand, you can create a subset of the records that have already met the selection criteria specified on the SELECT subcommand. ("SELECT subcommand" on page 130 describes SELECT/EVENT groups in more detail.)

The EVENT subcommand applies to PROCESS records only.

Keep in mind that the report is compiled by the number of records processed, which is determined by the SELECT subcommand, not just the records listed, which is determined by the EVENT subcommand. Therefore, it is possible for a report to have record totals in it that do not match the number of records for which you have set the criteria. The report totals list all the records that it processed in creating the report.

The syntax of the EVENT subcommand:

```
{EVENT}          event-name
{EV  }

                [EVQUAL(value-list...)]
                [CLASS(name-list...)]
                [NAME(name-list...)]
                [DSQUAL(name-list...)]
                [INTENT( [ALTER] [CONTROL] [UPDATE] ]
                [      [READ]  [NONE]  )           ]
                [ALLOWED( [ALTER] [CONTROL] [UPDATE] ]
                [      [READ]  [NONE]  )           ]
                [NEWNAME(name-list...)]
                [NEWSQUAL(name-list...)]
                [      {begin-number:end-number} ]
                [ LEVEL( {      } ) ]
                [      {number-list...      } ]
```


event-name

specifies one of the following valid event names:

LOGON

TSO logon or batch job initiation

ACCESS

Access to a RACF-protected resource

ADDVOL

Add a volume to a multivolume data set or tape volume set

RENAME

Rename a data set, SFS file, or SFS directory

DELETE

Delete a resource

DELVOL

Delete one volume of a multivolume data set or tape volume set

DEFINE

Define a resource

ALLSVC

All of the preceding functions (ACCESS, ADDVOL, RENAME, DELETE, DELVOL, and DEFINE)

ADDSD

ADDSD command

ADDGROUP

ADDGROUP command

ADDUSER

ADDUSER command

ALTDSD

ALTDSD command

ALTGROUP

ALTGROUP command

ALTUSER

ALTUSER command

CONNECT

CONNECT command

DELDSD

DELDSD command

DELGROUP

DELGROUP command

DELUSER

DELUSER command

PASSWORD

PASSWORD command

PERMIT

PERMIT command

RALTER

RALTER command

RDEFINE
RDEFINE command

RDELETE
RDELETE command

REMOVE
REMOVE command

RVARY
RVARY command

SETROPTS
SETROPTS command

ALLCOMMAND
All of the preceding RACF commands (ADDSD through SETROPTS)

APPCLU
Partner LU verification through use of APPCLU profile.

GENERAL
General purpose auditing

Not all of the EVENT subcommand operands are valid with certain event names.

EVQUAL(value-list...)
specifies a list of event qualifiers to be selected.

CLASS(class-name...)
specifies a list of resource class names to be selected. Only the DATASET class and class names found in the class descriptor table are valid.

NAME(name-list...)
specifies a list of resource names to be selected. In the NAME field, you must specify a fully qualified data set name, *not* a profile name for RACF SVC events (ACCESS, ADDVOL, RENAME, DELETE, DELVOL, DEFINE, ALLSVC). However, you must specify a profile name, *not* a fully qualified data set name, in the NAME field for RACF command events (ADDSD, ALTDSD, DELDSD, PERMIT, RALTER, RDEFINE, RDELETE, ALLCOMMAND).

To select specific data sets, you must specify fully qualified dataset names in the 'name-list'. Also, if a dataset has been renamed and you want to use this operand to select the old dataset name, you must specify the fully qualified, old data set name in the 'name-list'. This operand is not valid with the LOGON event name. You can specify generic names if you are looking for commands issued against that profile.

DSQUAL(name-list...)
specifies a list of dataset qualifiers to be selected. Valid dataset qualifiers are any user IDs or group names used as the high-level qualifier of a dataset name or any qualifiers supplied by the ICHRSMFE installation-wide exit routine. If a data set has been renamed and you want to use this operand to select the old dataset name, you must specify the qualifier of the old dataset name in the 'name-list'.

To obtain records that are pertinent solely to the dataset class, you must also specify CLASS(DATASET); otherwise, you receive records for all valid classes.

INTENT
specifies a list of intended access authorities to be selected. An intended access authority is the minimum authority needed by a user to access a particular resource (not the actual authority held by the user). The valid intended access

authorities are ALTER, CONTROL, UPDATE, READ, and NONE. The INTENT operand is valid only with the ACCESS event name.

ALLOWED

specifies a list of allowed access authorities to be selected. An allowed access authority is the actual authority held by the user requesting access to a particular resource (not the minimum authority needed by the user to access that resource). The valid, allowed access authorities are ALTER, CONTROL, UPDATE, READ, and NONE. The ALLOWED operand is valid only with either the ACCESS or the ADDVOL event names.

NEWNAME(name-list...)

specifies a list of new, fully qualified resource names to be selected. This operand is valid only with the RENAME event name.

NEWDSQUAL(name-list...)

specifies a list of qualifiers for new dataset or generic names to be selected. Valid qualifiers are any user IDs or group names used as the high-level qualifier of a dataset name or any qualifiers supplied by the ICHRSMFE installation-wide exit routine. This operand is valid only with the RENAME event name.

LEVEL(begin-number:end-number) or LEVEL(number-list)

specifies a range (in ascending order) or a list of resource levels to be selected.

The meaning of the level indicator is set by your installation with the ADDSD, ALTDSD, RDEFINE, and RALTER commands. See *z/OS Security Server RACF Command Language Reference* for more information about the LEVEL operand.

LIST subcommand

The LIST subcommand formats and prints a listing of each individual RACF SMF record (both PROCESS and STATUS) that passes the selection criteria specified on the SELECT and EVENT subcommands. On the LIST subcommand, you can specify the title, sort sequence, and format control for the listing. The RACF report writer processes only one LIST subcommand at a time; if you enter more than one, the RACF report writer recognizes only the last LIST subcommand that you have entered. (The RACF report writer does all processing after you enter the END command.)

If you want to execute a LIST subcommand more than once to produce your reports, you must run the report writer each time. If you use the same selection criteria for each LIST subcommand you run, use the SAVE operand on RACFRW to specify the work-data set that is to contain the selected, reformatted SMF records. In this way, you can avoid unnecessary processing each time you run the report writer.

The syntax of the LIST subcommand:

```
{LIST}          [TITLE('q-string')]
{L   }

                [SORT( [DATE] [TIME] [SYSID]      ]
                [      [USER] [GROUP] [EVENT]     ]
                [      [EVQUAL] [TYPE] [NAME]      ]
                [      [CLASS] [TERMINAL] [JOBID]  ]
                [      [OWNER] [SECLABEL]         ]
                [      [APPLAUDIT])               ]
```

[{ASCEND }]
 [{DESCEND}]
 [NEWPAGE]

TITLE('q-string')

specifies a string of up to 132 characters, enclosed in single quotation marks, to be used as the heading for each page of this particular listing. If you omit this operand but specify a default heading in the TITLE operand of the RACFRW command, the default heading appears on each page of the listing. If you omit both this operand and the RACFRW TITLE operand, no heading at all appears on the listing.

SORT(field-list)

specifies the fields of the input record (a reformatted RACF SMF record) that are to be used for sorting. If you specify the LIST subcommand without specifying the SORT operand, the RACF report writer sorts the records by RCDTYPE, at offset 5(5) in the reformatted SMF record, with STATUS records preceding PROCESS records. If you specify SORT operand values, the records are then further sorted within the STATUS and PROCESS groups by the fields that you specify on the SORT operand.

The sequence in which you specify the SORT operands determines the sequence in which the RACF report writer sorts the records. For example, specifying SORT(OWNER GROUP USER DATE TIME) causes the RACF report writer to sort according to the profile owner first, then the group name, then the user name. If you omit the SORT operand, the order in which the records were written to SMF is not necessarily the order in which the records appear in the output listing, unless you have specified EQUALS in the SORTEQU field of the installation-replaceable module (ICHRSMFI).

The following table describes the operands you can use to select a sort sequence. Even though these operands apply only to process records, specifying them does not affect the order of status records.

OPERAND	DESCRIPTION
DATE	Julian date (YYDDDF) that the job entered the system
TIME	Time of day (HHMMSSSTH)
SYSID	System identifier
USER	User (job) names
GROUP	Group (step) names
EVENT	Security-event codes
EVQUAL	Security-event code qualifiers
TYPE	Event types: 1 = JOB/LOGON events 2 = SVC events 3 = command events
NAME	Names of resources within event types: user ID for JOB/LOGON events RESOURCE NAME for SVC and command events
CLASS	Resource class names
TERMINAL	Terminal ID
JOBID	Job ID from SMF job management record
OWNER	Owner of the resource
SECLABEL	Security label
APPLAUDIT	APPLAUDIT key 8-byte key linking records of APPC/MVS transactions

ASCEND

specifies that the fields identified by the DATE and TIME operands are to be sorted in ascending order. If you omit the DATE and TIME operands, this operand is ignored.

ASCEND is the default value.

DESCEND

specifies that the fields identified by the DATE and TIME operands are to be sorted in descending order. If you omit both the DATE and TIME operands, this operand is ignored.

NEWPAGE

specifies that the listing is to start printing on a new page whenever the value in the major (first) sort field changes. If you omit the SORT operand, this operand is ignored.

SUMMARY subcommand

The SUMMARY subcommand causes the RACF report writer to format and print reports that summarize the information in the RACF SMF records that meet the selection criteria on the SELECT and EVENT subcommands.

Using the SUMMARY subcommand, you can request reports that summarize the following:

- Group activity
- User activity
- Resource activity
- Security-event activity
- RACF command activity
- Owner activity
- Group activity broken down by resource
- User activity broken down by resource
- Resource activity broken down by user
- Resource activity broken down by group
- Resource activity broken down by security event
- Security event activity broken down by resource
- RACF command activity broken down by user
- RACF command activity broken down by group
- RACF command activity broken down by resource
- Owner activity broken down by resource.

On a SUMMARY subcommand, you can specify only one of the activities mentioned in the preceding list. You can, however, enter as many as 16 different SUMMARY subcommands for each RACFRW command. You can thus request reports of all possible activities in one run of the RACF report writer. (Note that, if you accidentally enter more than one SUMMARY subcommand for the same type of activity, it does not cause an error; the RACF report writer recognizes only the last one.) The order in which you enter the SUMMARY subcommands is the order in which the summary reports are printed.

The syntax of the SUMMARY subcommand:

```
{SUMMARY}      name1  [BY(name2)]
{SUM          }

                [ {VIOLATIONS} ]
                [ {SUCCESES  } ]
                [ {WARNINGS  } ]

                [NEWPAGE]

                [TITLE('q-string')]
```

name1

specifies the major field on which information is to be grouped and summarized. The valid values for name1 are: GROUP, USER, RESOURCE, EVENT, COMMAND, and OWNER.

BY(name2)

specifies a minor field within the major field on which information is to be grouped and summarized also. The valid values for name2 are: GROUP, USER, RESOURCE, and EVENT.

Note: Only the following single name and name1 [BY(name2)] combinations are valid:

Name	name1 [BY(name2)]
GROUP	RESOURCE BY(USER)
USER	RESOURCE BY(GROUP)
RESOURCE	RESOURCE BY(EVENT)
EVENT	EVENT BY(RESOURCE)
COMMAND	COMMAND BY(USER)
OWNER	COMMAND BY(RESOURCE)
GROUP BY(RESOURCE)	COMMAND BY(GROUP)
USER BY(RESOURCE)	OWNER BY(RESOURCE)

VIOLATIONS

specifies that only information about access violations is to be included in the summary.

SUCCESES

specifies that only information about successful access attempts is to be included in the summary. If you omit VIOLATIONS, SUCCESES, and WARNING, the summary includes information for both access violations and successful access attempts.

WARNINGS

specifies that only accesses that were successful only because WARNING mode was in effect are to be included in the summary. The information appears under the WARNINGS heading.

If you do not specify VIOLATIONS, SUCCESES, or WARNINGS, the report summarizes all access attempts.

NEWPAGE

specifies that the summary report is to start printing on a new page whenever the value in name1 changes. NEWPAGE is valid only when BY(name2) is specified.

TITLE('q-string')

specifies a string of up to 132 characters, enclosed in single quotation marks, to be used as the heading for each page of this particular summary report. If you

omit this operand but specify a default heading in the TITLE operand of the RACFRW command, the default heading appears on each page of the summary report. If you omit both this operand and the RACFRW TITLE operand, no heading at all appears on the summary report.

END subcommand

The END subcommand terminates subcommand mode. All report-generation processing is done after you enter the END subcommand.

The syntax of the END subcommand:

```
END
```

Using the RACF report writer

Because of variations from one installation to another, it is not possible to identify all of the ways an auditor might use the RACF report writer. The following list, however, identifies some possibilities:

- “Monitoring password violation levels”
- “Monitoring access attempts in WARNING mode” on page 144
- “Monitoring access violations” on page 145
- “Monitoring the use of RACF commands” on page 146
- “Monitoring specific users” on page 146
- “Monitoring SPECIAL users” on page 147
- “Monitoring OPERATIONS users” on page 147
- “Monitoring failed accesses to resources protected by a security level” on page 147
- “Monitoring accesses to resources protected by a security label” on page 148.

The following detailed descriptions of these tasks include brief examples of the report writer command and subcommands needed for each. (In the examples, lowercase entries can be modified to suit the needs of your installation.) For sample reports, see “Sample reports” on page 152.

Monitoring password violation levels

Monitoring password violation levels enables you to:

- Determine how effectively new RACF users are coping with the LOGON process
- Determine if the number of password violations stabilizes over time
- Determine where (at which terminals) these password violations are occurring.

To obtain a report that describes password violations, you can use the following command and subcommands:

```
RACFRW GENSUM...  
SELECT PROCESS  
EVENT LOGON EVQUAL(1)  
LIST ...  
END
```

Results

These subcommands create a general summary report and a listing of the selected process records. (See Figure 36 on page 157 and Figure 38 on page 159 for samples of the general summary report and listings of selected process records.)

The total number of job or logon violations in the general summary report includes all types of violations (invalid password, invalid group, invalid OIDCARD, and invalid terminal). Because the EVENT subcommand causes the RACF report writer to select only those process records that describe an invalid password, you can use the number of process records selected to determine the percentage of password violations. If, for example, the number of process records selected is 13 and the total number of job or logon attempts is 393, you can compute the percentage of password violations by dividing 13 by 393. In this particular example, the value is 3.3%.

The violation percentage is a useful number to record and track over time. As users become more familiar with using their user ID and password, this percentage should tend to stabilize at a relatively low level.

You can look at the terminal name in the listing of process records to determine where persistent violations are originating. The records selected are record types 20, 30, and 80 (process records) with an event code of 1 for job initiation or TSO logon. (See Figure 33 on page 154 for a list of RACF events and their qualifiers.)

Monitoring access attempts in WARNING mode

Your installation may choose to use warning mode during the initial implementation of RACF. During this period, resource profiles contain a warning indicator (specified when the owner creates or later changes the profile). When the warning indicator is set, RACF allows all requesters to access the resource, and, if the requester would not otherwise be allowed access, RACF sends a message to the requester. Logging occurs at the owner-specified access type and level.

If the owner of a resource has specified in the profile one of the following:

- AUDIT(FAILURE(READ))
- AUDIT(ALL(READ)) (or the defaults for these are in effect)

or if you, as auditor, specify one of the following:

- GLOBALAUDIT (FAILURE(READ))
- GLOBALAUDIT (ALL(READ))

RACF logs each access to the resource, and you can use the RACF report writer to provide a list of the accesses RACF allowed only because the warning indicator was set.

Using the warning indicator can help your installation to migrate gradually to RACF. Checking the requesters and resources in the report-writer listing can enable you to develop access lists without disrupting authorized work and without the immediate need to write and test a RACF exit routine.

As the auditor, however, you must be aware that if your installation sets the warning indicator in a resource profile any requester can access the resource. You should verify that the profile for a highly classified resource (such as payroll or business-planning data) does not contain the warning indicator.

To obtain a list of the profiles in a particular class that have the warning indicator set, you can issue the RACF SEARCH command with the WARNING operand:

```
SEARCH CLASS(class-name) WARNING
```

For example, to list the profiles in the TERMINAL class that contain the warning indicator, enter:

```
SEARCH CLASS(TERMINAL) WARNING
```

To obtain a report of accesses granted only because the warning indicator was set, you can use the following command and subcommands:

```
RACFRW ...  
  SELECT PROCESS WARNINGS  
  LIST ...  
END
```

Results

These subcommands produce a listing of the selected process records. The records selected are those that contain an event code of 2 for resource access and a qualifier from the following table.

EVENT NUMBER	DESCRIPTION
--------------	-------------

- | | |
|----|--|
| 3 | Warning issued because of access. |
| 5 | Warning issued because of PROTECTALL. |
| 8 | Warning issued because of missing security label from job, user, or profile. |
| 9 | Warning issued because of insufficient security label authority. |
| 10 | Warning issued because data set is not cataloged. |
| 13 | Warning issued because of insufficient CATEGORY/SECLEVEL. |

The WARNING indicator is also set in records for the following events: LOGON, RENAME, DEFINE.

Monitoring access violations

When warning mode is in effect, and during normal operation of RACF, it is essential to your job as an auditor that you be able to monitor access violations. RACF detects and logs an access violation when it denies a user access to a resource because that user is not authorized to access the resource. An access violation is, therefore, a symptom that someone either does not understand their role as a RACF user or is trying to bypass RACF protection. You can use a report of access violations to identify such users and to help your installation identify when it may need to change access lists or universal access codes (UACCs).

You can request the report for data set violations and for violations in any of the classes identified in the class descriptor table.

To obtain an access violation report, you can use the following command and subcommands with the resource classes for which you want information:

```
RACFRW ...  
  LIST ...  
  SELECT PROCESS
```

```

EVENT ACCESS EVQUAL(1) CLASS(a valid resource class,...,
    a valid resource class)
EVENT LOGON EVQUAL(4)
END

```

Results

These subcommands create a listing of all process records that meet the criteria set in the EVENT subcommands. The EVENT ACCESS subcommand selects all process records that contain access violations for the specified classes (an event code of 2 and an event qualifier of 1). The EVENT LOGON subcommand expands the scope of the report to include all user attempts to log on from a terminal or console the user is not authorized to use (an event code of 1 and an event qualifier of 4).

Monitoring the use of RACF commands

In any installation, the security administrator is probably the most frequent user of RACF commands. Occasionally, users without any privileged attributes may enter ADDSD, PERMIT, or RDEFINE, or another, similar command against one of their resources; however, some users may try to use the whole range of RACF commands. Unless the user is authorized, RACF does not execute the command. Each unauthorized attempt to use a RACF command, however, represents a potential security violation, an event that you should know about. You monitor the use of commands with the command-summary report.

To obtain a command-summary report, you can use the following command and subcommand:

```

RACFRW ...
SUMMARY COMMAND BY (USER)
END

```

A sample command-by-user summary report appears in Figure 51 on page 169.

If you detect certain users making persistent, unauthorized use of RACF commands, you can extract the details of the commands used and the resources involved. To obtain details of any command violations logged for specific users, use the following command and subcommands:

```

RACFRW ...
SELECT VIOLATIONS USER(userid(s) ...)
LIST ...
END

```

Where *userid(s)* is the ID of the user making unauthorized use of RACF commands. Note that RACF does not automatically log the events that these reports describe. To obtain meaningful data, you must direct RACF to log the activities of specific users or command violations or both. The reports are useful only after RACF has logged the events for the time interval that is meaningful to you. See “Monitoring specific users,” “Monitoring SPECIAL users” on page 147, and “Monitoring OPERATIONS users” on page 147 for related information.

Monitoring specific users

If you have directed RACF, either through the UAUDIT operand on the ALTUSER command or the corresponding ISPF panel, to log the RACF-related activities of one or more specific users, you can use the report writer to obtain a listing of the activities of these users.

To obtain a listing of all records RACF has logged because you requested auditing of one or more specific users, you can use the following command and subcommands:

```
RACFRW ...  
  SELECT PROCESS REASON(USER) ...  
  LIST ...  
END
```

Monitoring SPECIAL users

If you have directed RACF, either through the SAUDIT operand on the SETROPTS command or the corresponding ISPF panel, to log the RACF-related activities of SPECIAL or group-SPECIAL users, you can use the report writer to obtain a listing of the activities of these users.

To obtain a listing of all records RACF has logged because you requested auditing of SPECIAL or group-SPECIAL users or because the command required the SPECIAL or group-SPECIAL attribute and the user had it, you can use the following command and subcommands:

```
RACFRW ...  
  SELECT PROCESS AUTHORITY(SPECIAL)  
  LIST ...  
END
```

Monitoring OPERATIONS users

The OPERATIONS and group-OPERATIONS attributes are very powerful. OPERATIONS allows a user access to almost all resources. Group-OPERATIONS allows a user access to almost all resources within the scope of the group and its subgroups. (The only resources not accessible to the OPERATIONS or group-OPERATIONS user are those that have been explicitly barred by placing the OPERATIONS user in the access list of a resource with an access level of NONE at either the user ID level or the group level.) Therefore, you should carefully monitor the activities of these users to ensure that all accesses to installation resources are for valid reasons.

To obtain a report of the activities of OPERATIONS and group-OPERATIONS users, you can use the following command and subcommand:

```
RACFRW ...  
  LIST ...  
  SELECT PROCESS AUTHORITY(OPERATIONS)  
END
```

Note: RACF logs the activities of users with the OPERATIONS and group-OPERATIONS attributes if the following are true:

- The SETROPTS OPERAUDIT is in effect.
- The access to the resource was successful because the user had the OPERATIONS or group-OPERATIONS attribute.

Monitoring failed accesses to resources protected by a security level

If you have directed RACF, through the SECLEVELAUDIT operand on the SETROPTS command or on the corresponding ISPF panel, to log accesses to resources that are protected by a security level, you can use the report writer to

obtain a listing of any access attempts that have failed because the user did not have the sufficient security classification to access the resource.

When security-level auditing is in effect, RACF logs all attempts to access any resource protected by a given security level (such as "confidential") or higher. Therefore, you can create a report to list access violations to those protected resources and determine which users are attempting to access sensitive information at your installation.

To obtain a report of unauthorized access attempts to resources with a security-level classification, you can use the following command and subcommands:

```
RACFRW
  SELECT PROCESS REASON(SECAUDIT)
  EVENT ACCESS EVQUAL(6) CLASS(a valid resource class,. . .,
    a valid resource class)
  LIST
END
```

Result

These subcommands create a listing of all process records that have been logged because security-level auditing was in effect (REASON(SECAUDIT)) and meet the criteria set in the EVENT ACCESS subcommand (event code 2). The EVENT subcommand selects all failed attempts (event qualifier 6) to access any resource within the resource class that has a security level equal to or higher than the level specified on the SECLEVELAUDIT operand of the SETROPTS command or on the corresponding ISPF panel.

Monitoring accesses to resources protected by a security label

If you have directed RACF, through the SECLABELAUDIT operand on the SETROPTS command or on the corresponding ISPF panel, to log accesses to resources that are protected by a security label according to the audit options in the SECLABEL profile, you can use the report writer to obtain a listing of all attempts to access the resource.

When the SECLABELAUDIT option is in effect, RACF logs accesses to resources by SECLABEL. Therefore, you can create a report to list attempts to access those protected resources and determine which users are attempting to access sensitive information at your installation.

To obtain a report of attempts to access resources with a security label, you can use the following command and subcommands:

```
RACFRW
  SELECT PROCESS REASON(SECLABELAUDIT)
  EVENT ACCESS
  LIST
END
```

Result

These subcommands create a listing of all process records that have been logged because the security-label auditing option was in effect (REASON(SECLABELAUDIT)) and meet the criteria set in the EVENT subcommand ACCESS (event code 2).

RACF report writer examples

This section gives some examples of how to use the RACF report writer command and subcommands to produce various reports.

The first five examples show how to obtain single reports; however, to create all the reports that you require at your installation, you may need to execute the RACF report writer more than once.

An execution of the RACF report writer consists of the RACFRW command, report definition subcommands, and the END subcommand. Example 5 shows how the report writer executed a series of subcommands to produce multiple reports that you did not intend to produce; example 6 shows how you can correct the subcommands to produce the number of reports you want.

Example 1—Obtaining a report for all RACF SMF records

To obtain a report of all RACF SMF records, listed in the order read from the input file, and a general summary report, showing overall RACF-related system activity, enter:

```
RACFRW TITLE('BIG LISTING') GENSUM
LIST
END
```

Example 2—Obtaining a report for all MVS jobs run by users not defined to RACF

To obtain a report of all batch jobs that are not associated with RACF or a RACF-defined user, or all jobs run by TSO users, or started tasks not defined to RACF, enter:

```
RACFRW
SELECT NOUSER PROCESS
LIST TITLE('JOB LIST REPORT') SORT(USER) NEWPAGE
```

In the example, RACF selects only those process records that meet the criteria and sorts by job name.

To obtain a summary of these jobs, enter:

```
SUMMARY RESOURCE TITLE('JOB SUMMARY REPORT')
END
```

Example 3—Obtaining a report for data set violations

To obtain a report of all violations against data sets owned by USERA (USERA is the high-level qualifier of the data-set name) in January 1989, sorted in date and time sequence, enter:

```
RACFRW TITLE('USERA DATASETS LIST REPORT')
SELECT VIOLATIONS DATE(89001:89031)
EVENT ALLSVC CLASS(DATASET) DSQUAL(USERA)
EVENT ALLCOMMAND CLASS(DATASET) DSQUAL(USERA)
LIST SORT(DATE TIME)
```

To obtain a summary of this activity, enter:

```
SUMMARY RESOURCE BY(USER) TITLE('USERA DATA SETS SUMMARY REPORT')
```

Example 4—Obtaining a report for data set activity by job, system, and user

To obtain a report on data set activity by (a) jobs A and B on system 308A and (b) users C and D on system 308B, enter:

```
RACFRW
SELECT JOB(A B) NOUSER SYSID(308A)
EVENT ALLSVC CLASS(DATASET)
EVENT ALLCOMMAND CLASS(DATASET)
SELECT USER(C D) NOJOB SYSID(308B)
EVENT ALLSVC CLASS(DATASET)
EVENT ALLCOMMAND CLASS(DATASET)
LIST TITLE('SELECTED DATA SET ACTIVITY REPORT') SORT(SYSID)
END
```

Example 5—Obtaining multiple reports the wrong way

Situation

Assume you need to produce the following separate reports:

- A detailed listing of all access violations, sorted by user
- A resource-by-user summary report, with totals for access violations only
- A listing of all successful accesses, sorted by date and time
- A resource-by-user summary report, with totals for successful accesses only.

You must produce these *four separate* reports because each report is to be distributed to four different people, each of whom is entitled to see only the information on one report.

Assume that you enter:

- (1) RACFRW
- (2) SELECT VIOLATIONS
- (3) LIST TITLE('ACCESS VIOLATIONS LIST REPORT') SORT(USER)
- (4) SUMMARY RESOURCE BY(USER) TITLE ('ACCESS VIOLATIONS SUMMARY REPORT')
- (5) SELECT SUCCESSES
- (6) LIST TITLE('ACCESS SUCCESS LIST REPORT') SORT(DATE TIME)
- (7) SUMMARY RESOURCE BY(USER) TITLE('ACCESS SUCCESS SUMMARY REPORT')
- (8) END

Result

Instead of receiving the four requested reports, you receive *two* reports:

- A list report of all violations and successes, sorted by date and time

- A summary report of resources-by-user, with both violations and successful accesses.

How RACF executed

Here is what happened:

- **RACF record selection**

You intended to first select, list, and summarize only violations from the SMF input file (statements 2, 3, and 4). Second, you wanted to select, list, and summarize only successful accesses (statements 5, 6, and 7), and finally, you wanted to produce two summary reports, one for access violations and one for access successes (statements 4 and 7).

However, the RACF report writer does not execute in that sequence. RACF first selects records based on *all* the SELECT and EVENT subcommands entered between the RACFRW command and the END subcommand. Only after this selection process is complete are any of the requested reports produced. In this example, the RACF report writer checked each record from the input file to see whether it was either an access violation (statement 2) or a successful access (statement 5). Because all of the SMF records met at least one of these conditions, the RACF report writer selected all of the records for further processing.

- **RACF LIST function**

The RACF report writer next produced a single list report (statement 6). RACF ignored the first LIST subcommand (statement 3) because only one LIST subcommand, the last one entered (statement 6), is valid for each execution of the RACF report writer. The report that was produced listed by date and time all the records selected (both access violations and successful accesses) as specified in statement 6.

- **RACF SUMMARY report**

Next, the RACF report writer produced a single summary report (statement 7). Because the SUMMARY subcommand in statement 4 is the same as that in statement 7, RACF ignored the first SUMMARY subcommand and produced one summary report. If you enter identical SUMMARY subcommands between RACFRW and END, RACF only uses the last subcommand and produces one summary report.

Thus, the single summary report for this example produced totals for all the records selected (both access violations and successful accesses).

Example 6—Obtaining multiple reports the right way

To produce the four listings that you intended, enter two separate RACFRW commands:

- (1) RACFRW
 SELECT VIOLATIONS
 LIST TITLE('ACCESS VIOLATIONS LIST REPORT') SORT(USER)
 SUMMARY RESOURCE BY(USER) TITLE ('ACCESS VIOLATIONS
 SUMMARY REPORT')
 END
- (2) RACFRW
 SELECT SUCCESSES


```

LIST TITLE('ACCESS SUCCESS LIST REPORT') SORT(DATE TIME)
SUMMARY RESOURCE BY(USER) TITLE ('ACCESS SUCCESS SUMMARY
REPORT')
END

```

Note: RACF interprets each RACFRW command separately and produces the four reports. To ensure you get the reports you want:

1. If you want to store the results in a GDG data set, use DISP=MOD on your JCL to prevent the results of the second RACFRW operation from writing over the results of the first.
2. After the first SELECT/LIST/SUMMARY subcommands (for RACFRW in statement 1), be sure to enter **END**.
3. Run the RACFRW command again (statement 2) for the second SELECT/LIST/SUMMARY subcommands and enter **END**.

Sample reports

This section includes examples of the various reports that you can request the RACF report writer to generate. Review each sample report to determine its usefulness to your particular installation.

The following list summarizes the sample reports and the command or subcommand you issue to request the report:

Figure	Report	Command/Subcommand Issued
4-11	Summary Activity Report	From SMF
4-12	Standard Header Page	Each time you invoke the RACF report writer, it produces a standard header page that lists the subcommands that you entered and describes the meanings of the event and event qualifier values used in the reports.
4-13	General Summary	RACFRW GENSUM
4-14	Listing of Status Records (types 80 and 81)	LIST (see Note)
4-15	Listing of Process Records (types 20, 30, 80 and 83)	LIST (see Note)
4-16	Short User Summary	SUMMARY USER
4-17	Short Group Summary	SUMMARY GROUP
4-18	Short Resource Summary	SUMMARY RESOURCE
4-19	Short Command Summary	SUMMARY COMMAND
4-20	Short Event Summary	SUMMARY EVENT
4-21	Short Owner Summary	SUMMARY OWNER
4-22	User by Resource Summary	SUMMARY USER BY(RESOURCE)
4-23	Group by Resource Summary	SUMMARY GROUP BY(RESOURCE)
4-24	Resource by User Summary	SUMMARY RESOURCE BY(USER)
4-25	Resource by Group Summary	SUMMARY RESOURCE BY(GROUP)
4-26	Resource by Event Summary	SUMMARY RESOURCE BY(EVENT)
4-30	Event by Resource Summary	SUMMARY EVENT BY(RESOURCE)
4-28	Command by User Summary	SUMMARY COMMAND BY(USER)
4-29	Command by Group Summary	SUMMARY COMMAND BY(GROUP)
4-30	Command by Resource Summary	SUMMARY COMMAND BY(RESOURCE)

Figure Report	Command/Subcommand Issued
4-31 Owner by Resource Summary	SUMMARY OWNER BY(RESOURCE)

Note: A single LIST subcommand produces both the listing of status records and the listing of process records.

An explanation of the standard header page of the report is given in "Event code qualifiers" on page 69. It documents *why* the event code qualifiers were set.

SUMMARY ACTIVITY REPORT								
START DATE-TIME	07/29/89-13:18:18			END DATE-TIME	08/06/89-10:15:36			
RECORD TYPE	RECORDS READ	PERCENT OF TOTAL	AVG. RECORD LENGTH	MIN. RECORD LENGTH	MAX. RECORD LENGTH	RECORDS WRITTEN		
0	2	.55 %	35.00	35	35	0		
2	0					1		
3	0					1		
4	41	11.33 %	251.48	207	263	0		
5	24	6.63 %	143.70	137	144	0		
20	52	14.36 %	94.23	91	98	52		
30	133	36.74 %	577.61	244	2,174	133		
80	108	29.83 %	450.78	80	1,685	108		
81	2	.55 %	756.00	756	756	2		
TOTAL	362	100 %	402.00	35	2,174	297		
NUMBER OF RECORDS IN ERROR		0						

Figure 32. Summary Activity Report from SMF

90.053 13:51:40
COMMAND GROUP ENTERED -
RACFRW GENSUM
LIST
END

RACF REPORT

```
EVENT/QUALIFIER KEY -----  
EVENT  QUALIFIER  MEANING  
1  
0      JOB INITIATION / TSO LOGON/LOGOFF  
0      SUCCESSFUL INITIATION  
1      INVALID PASSWORD  
2      INVALID GROUP  
3      INVALID OIACARD  
4      INVALID TERMINAL/CONSOLE  
5      INVALID APPLICATION  
6      REVOKED USERID ATTEMPTING ACCESS  
7      USERID AUTOMATICALLY REVOKED  
8      SUCCESSFUL TERMINATION  
9      UNDEFINED USERID  
10     INSUFFICIENT SECURITY LABEL AUTHORITY  
11     NOT AUTHORIZED TO SECURITY LABEL  
12     SUCCESSFUL RACINIT INITIATION  
13     SUCCESSFUL RACINIT DELETE  
14     SYSTEM NOW REQUIRES MORE AUTHORITY  
15     REMOTE JOB ENTRY - JOB NOT AUTHORIZED  
16     SURROGAT CLASS IS INACTIVE  
17     SUBMITTER IS NOT AUTHORIZED BY USER  
18     SUBMITTER IS NOT AUTHORIZED TO SECURITY LABEL  
19     USER IS NOT AUTHORIZED TO JOB  
20     WARNING - INSUFFICIENT SECURITY LABEL AUTHORITY  
21     WARNING - SECURITY LABEL MISSING FROM JOB, USER, OR PROFI  
22     WARNING - NOT AUTHORIZED TO SECURITY LABEL  
23     SECURITY LABELS NOT COMPATIBLE  
24     WARNING - SECURITY LABELS NOT COMPATIBLE  
25     CURRENT PASSWORD HAS EXPIRED  
26     INVALID NEW PASSWORD  
27     VERIFICATION FAILED BY INSTALLATION  
28     GROUP ACCESS HAS BEEN REVOKED  
29     OIACARD IS REQUIRED  
30     NETWORK JOB ENTRY - JOB NOT AUTHORIZED  
31     WARNING - UNKNOWN USER FROM TRUSTED NODE PROPAGATED  
32     SUCCESSFUL INITIATION USING PASSTICKET  
33     INDICATES ATTEMPTED REPLAY OF PASSTICKET
```

Figure 33. Standard Header Page (1/3)

```

2      RESOURCE ACCESS
      0      SUCCESSFUL ACCESS
      1      INSUFFICIENT AUTHORITY
      2      PROFILE NOT FOUND - RACFIND SPECIFIED ON MACRO
      3      ACCESS PERMITTED DUE TO WARNING
      4      FAILED DUE TO PROTECTALL
      5      WARNING ISSUED DUE TO PROTECTALL
      6      INSUFFICIENT CATEGORY/SECLEVEL
      7      INSUFFICIENT SECURITY LABEL AUTHORITY
      8      WARNING - SECURITY LABEL MISSING FROM JOB, USER, OR PROFILE
      9      WARNING - INSUFFICIENT SECURITY LABEL AUTHORITY
     10      WARNING - DATA SET NOT CATALOGUED
     11      DATA SET NOT CATALOGUED
     12      PROFILE NOT FOUND - REQUIRED FOR AUTHORITY CHECKING
     13      WARNING: INSUFFICIENT CATEGORY/SECLEVEL
3      ADDVOL/CHGVOL
      0      SUCCESSFUL PROCESSING OF NEW VOLUME
      1      INSUFFICIENT AUTHORITY
      2      INSUFFICIENT SECURITY LABEL AUTHORITY
      3      LESS SPECIFIC PROFILE EXISTS WITH DIFFERENT SECLABEL
4      RENAME RESOURCE
      0      SUCCESSFUL RENAME
      1      INVALID GROUP
      2      USER NOT IN GROUP
      3      INSUFFICIENT AUTHORITY
      4      RESOURCE NAME ALREADY DEFINED
      5      USER NOT DEFINED TO RACF
      6      RESOURCE NOT PROTECTED
      7      WARNING - RESOURCE NOT PROTECTED
      8      USER IN SECOND QUALIFIER IS NOT RACF DEFINED
      9      LESS SPECIFIC PROFILE EXISTS WITH DIFFERENT SECLABEL
     10      INSUFFICIENT SECURITY LABEL AUTHORITY
     11      RESOURCE NOT PROTECTED BY SECURITY LABEL
     12      NEW NAME NOT PROTECTED BY SECURITY LABEL
     13      NEW SECLABEL MUST DOMINATE OLD SECLABEL
     14      WARNING - INSUFFICIENT SECURITY LABEL AUTHORITY
     15      WARNING - RESOURCE NOT PROTECTED BY SECURITY LABEL
     16      WARNING - NEW NAME NOT PROTECTED BY SECURITY LABEL
     17      WARNING - NEW SECLABEL MUST DOMINATE OLD SECLABEL
5      DELETE RESOURCE
      0      SUCCESSFUL SCRATCH
      1      RESOURCE NOT FOUND
      2      INVALID VOLUME
6      DELETE ONE VOLUME OF A MULTIVOLUME RESOURCE
      0      SUCCESSFUL DELETION
7      DEFINE RESOURCE
      0      SUCCESSFUL DEFINITION
      1      GROUP UNDEFINED
      2      USER NOT IN GROUP
      3      INSUFFICIENT AUTHORITY
      4      RESOURCE NAME ALREADY DEFINED
      5      USER NOT DEFINED TO RACF
      6      RESOURCE NOT PROTECTED
      7      WARNING - RESOURCE NOT PROTECTED
      8      WARNING - SECURITY LABEL MISSING FROM JOB, USER, OR PROFILE
      9      WARNING - INSUFFICIENT SECURITY LABEL AUTHORITY
     10      USER IN SECOND QUALIFIER IS NOT RACF DEFINED
     11      INSUFFICIENT SECURITY LABEL AUTHORITY
     12      LESS SPECIFIC PROFILE EXISTS WITH DIFFERENT SECLABEL

```

Figure 34. Standard Header Page (2/3)

90.053 13:51:40

RACF REPORT

```
8      ADDSD COMMAND
9      ADDGROUP COMMAND
10     ADDUSER COMMAND
11     ALTDSD COMMAND
12     ALTGROUP COMMAND
13     ALTUSER COMMAND
14     CONNECT COMMAND
15     DELDSD COMMAND
16     DELGROUP COMMAND
17     DELUSER COMMAND
18     PASSWORD COMMAND
19     PERMIT COMMAND
20     RALTER COMMAND
21     RDEFINE COMMAND
22     RDELETE COMMAND
23     REMOVE COMMAND
24     SETROPTS COMMAND
25     RVARY COMMAND
      0      NO VIOLATIONS DETECTED
      1      INSUFFICIENT AUTHORITY
      2      KEYWORD VIOLATIONS DETECTED
      3      SUCCESSFUL LISTING OF DATA SETS
      4      SYSTEM ERROR IN LISTING OF DATA SETS
26     APPCLU
      0      PARTNER VERIFICATION WAS SUCCESSFUL
      1      SESSION ESTABLISHED WITHOUT VERIFICATION
      2      LOCAL LU KEY WILL EXPIRE IN <= 5 DAYS
      3      PARTNER LU ACCESS HAS BEEN REVOKED
      4      PARTNER LU KEY DOES NOT MATCH THIS LU KEY
      5      SESSION TERMINATED FOR SECURITY REASON
      6      REQUIRED SESSION KEY NOT DEFINED
      7      POSSIBLE SECURITY ATTACK BY PARTNER LU
      8      SESSION KEY NOT DEFINED FOR PARTNER LU
      9      SESSION KEY NOT DEFINED FOR THIS LU
     10     SNA SECURITY RELATED PROTOCOL ERROR
     11     PROFILE CHANGE DURING VERIFICATION
     12     EXPIRED SESSION KEY
27     GENERAL
      0-99   GENERAL AUDIT RECORD WRITTEN
-REPORT KEY -----
.AN '*' PREFIXED TO A USER OR GROUP NAME INDICATES THE NAME IS ACTUALLY A JOB OR STEP NAME,
RESPECTIVELY
.THE PHRASE 'UNDEFINED USER' REFERS TO THOSE TSO LOGONS WHICH SPECIFIED USERIDS THAT WERE NOT
DEFINED TO RACF,
AND TO BATCH JOBS WHICH DID NOT SPECIFY THE 'USER=' OPERAND ON THEIR JOB STATEMENTS
.A '+' PREFIXED TO A RESOURCE NAME INDICATES THAT A GENERIC PROFILE WAS ACCESSED
.A '(G)' APPENDED TO A RESOURCE NAME MEANS THAT THE RESOURCE NAME IS GENERIC
.A '-' APPENDED TO A VMXEVT DESCRIPTION MEANS THAT THE EVENT CONTINUES ON THE NEXT LINE
```

Figure 35. Standard Header Page (3/3)

90.053 13:51:40

RACF REPORT - GENERAL SUMMARY

	READ	SELECTED	%-SELECTED				
STATUS RECORDS	49	49	100 %				
PROCESS RECORDS	126	126	100 %				
TOTAL PROCESS RECORDS FOR DEFINED USERS	125	125	99 % (OF ALL PROCESS RECORDS)				
TOTAL PROCESS RECORDS FOR UNDEFINED USERS	1	1	1 % (OF ALL PROCESS RECORDS)				
--- JOB / LOGON STATISTICS ---							
TOTAL JOB/LOGON/LOGOFF	19						
TOTAL JOB/LOGON SUCCESSES	4		21 % OF TOTAL ATTEMPTS				
TOTAL JOB/LOGON VIOLATIONS	10		53 % OF TOTAL ATTEMPTS				
TOTAL JOB/LOGON ATTEMPTS BY UNDEFINED USERS	1		5 % OF TOTAL ATTEMPTS				
TOTAL JOB/LOGON SUCCESSES BY UNDEFINED USERS	0		0 % OF TOTAL ATTEMPTS				
TOTAL JOB/LOGON VIOLATIONS BY UNDEFINED USERS	1		5 % OF TOTAL ATTEMPTS				
TOTAL JOB/LOGON SUCCESSFUL TERMINATION	5						
JOB/LOGON VIOLATIONS BY HOUR -							
0-1	1-2	2-3	3-4	4-5	5-6	6-7	7-8
0	0	0	0	0	0	0	0
8-9	9-10	10-11	11-12	12-13	13-14	14-15	15-16
0	0	0	0	10	0	0	0
16-17	17-18	18-19	19-20	20-21	21-22	22-23	23-24
0	0	0	0	0	0	0	0
--- RESOURCE STATISTICS ---							
TOTAL RESOURCE ACCESSES (ALL EVENTS)	45						
TOTAL RESOURCE ACCESS SUCCESSES	44		98 % OF TOTAL ACCESSES				
TOTAL RESOURCE ACCESS WARNINGS	0		0 % OF TOTAL ACCESSES				
TOTAL RESOURCE ACCESS VIOLATIONS	1		2 % OF TOTAL ACCESSES				
TOTAL RESOURCE ACCESSES (ALL EVENTS) BY UNDEFINED USERS	0		0 % OF TOTAL ACCESSES				
TOTAL RESOURCE ACCESS SUCCESSES BY UNDEFINED USERS	0		0 % OF TOTAL ACCESSES				
TOTAL RESOURCE ACCESS WARNINGS BY UNDEFINED USERS	0		0 % OF TOTAL ACCESSES				
TOTAL RESOURCE ACCESS VIOLATIONS BY UNDEFINED USERS	0		0 % OF TOTAL ACCESSES				
TOTAL RESOURCE ACCESSES USING GENERIC PROFILE	5		11 % OF TOTAL ACCESSES				
TOTAL RESOURCE ACCESS SUCCESSES USING GENERIC PROFILE	5		11 % OF TOTAL ACCESSES				
TOTAL RESOURCE ACCESS WARNINGS USING GENERIC PROFILE	0		0 % OF TOTAL ACCESSES				
TOTAL RESOURCE ACCESS VIOLATIONS USING GENERIC PROFILE	0		0 % OF TOTAL ACCESSES				
RESOURCE ACCESS VIOLATIONS BY HOUR -							
0-1	1-2	2-3	3-4	4-5	5-6	6-7	7-8
0	0	0	0	0	0	0	0
8-9	9-10	10-11	11-12	12-13	13-14	14-15	15-16
0	0	0	0	1	0	0	0
16-17	17-18	18-19	19-20	20-21	21-22	22-23	23-24
0	0	0	0	0	0	0	0

Figure 36. General Summary Report

```

90.053 13:51:40          RACF REPORT - LISTING OF STATUS RECORDS
DATE  TIME  SYSID MISC. OPTIONS  EXITS  CLASS  PROT  STAT  AUD  GEN  GCMD  GLBL  GLST  RLST  LOPT
90.053 12:17:41 R190  ORIGIN:  SETROPTS  DATASET  YES  YES  NO  YES  YES  YES
      TERMUACC: READ  USER  NO
      CMNDVIOL: YES  GROUP  NO
      LOGSPEC: YES  RVARSMBR  YES  NO  YES  YES  YES  DFLT
      RACINIT: STATS  RACFVARS  YES  NO  YES  YES  YES  YES  DFLT
      ADSP:  ACTIVE  SECLABEL  YES  NO  YES  YES  YES  DFLT
      REALDSN: NO  DASDVOL  NO  NO  YES  YES  YES  DFLT
      JES:  GDASDVOL  NO  NO  NO  DFLT
      BATCHALLRACF  TAPEVOL  YES  NO  YES  YES  YES  DFLT
      XBMAILRACF  TERMINAL  YES  NO  YES  YES  YES  DFLT
      EARLYVERIFY  GTERMINL  YES  NO  NO  DFLT
      APPL  NO  NO  NO  YES  YES  YES  DFLT
      TAPEDSN: NO  TIMS  NO  NO  YES  YES  YES  DFLT
      PROT-ALL: NO  GIMS  NO  NO  NO  DFLT
      PROGCTL: NO  AIMS  NO  NO  YES  YES  YES  DFLT
      OPERAUDIT:NO  TCICSTRN  NO  NO  YES  YES  YES  DFLT
      ERASE:  YES  GCICSTRN  NO  NO  NO  DFLT
      NOSECLEVEL  PCICSPSB  NO  NO  YES  YES  YES  DFLT
      ALL  QCICSPSB  NO  NO  NO  DFLT
      SECLEVELAUDITING INACTIVE  GLOBAL  NO  NO  NO  DFLT
      EGN:  INACTIVE  GMBR  NO  NO  YES  YES  YES  DFLT
      SESSIONINTERVAL 30  DSNR  NO  NO  YES  YES  YES  DFLT
      JES B1 SECURITY:  FACILITY  NO  NO  YES  YES  YES  DFLT
      NJEUSERID: UNKUSER  VMMDISK  NO  NO  YES  YES  YES  DFLT
      UNDEFINEDUSER: ++++++++  VMRDR  NO  NO  YES  YES  YES  DFLT
      DEFAULT LANGUAGE CODES:  SECDATA  NO  NO  NO  DFLT
      PRIMARY CODE: ENU  PROGRAM  NO  NO  NO  DFLT
      SECONDARY CODE: ENU  APPCLU  NO  NO  YES  YES  YES  DFLT
      APPLAUDIT: YES
      JESJOBS  YES  NO  NO  YES  YES  YES  DFLT
      JESINPUT  YES  NO  NO  YES  YES  YES  DFLT
      CONSOLE  YES  NO  NO  YES  YES  YES  YES  DFLT
      TEMPDSN  YES  NO  NO  YES  YES  YES  DFLT
      DIRAUTH  YES  NO  NO  YES  YES  YES  DFLT
      SURROGAT  YES  NO  NO  YES  YES  YES  DFLT
      NODMBR  YES  NO  NO  YES  YES  YES  DFLT
      NODES  YES  NO  NO  YES  YES  YES  YES  DFLT
      OTHER OPTIONS -
      'LIST OF GROUPS' ACCESS CHECKING IS ACTIVE
      SINGLE LEVEL NAMES NOT ALLOWED
      INTERVAL: 253 DAYS
      HISTORY: NONE
      REVOKE: NO
      WARNING: NONE
      INACTIVE: NO
      NO PASSWORD SYNTAX RULES
      SECURITY OPTIONS:
      SECLABELCONTROL: INACTIVE
      CATDSNS: INACTIVE
      MLQUIET: INACTIVE
      MLSTABLE: INACTIVE
      MLS: INACTIVE
      MLACTIVE: INACTIVE
      GENERICOWNER: INACTIVE
      SECLABELAUDIT: INACTIVE
      COMPATMODE: INACTIVE

```

Figure 37. Listing of Status Records

If the LRECL value specified is too small, the report output shown in Figure 37 contains the report heading and the following text:

```

**** STATUS RECORD BYPASSED; LRECL TOO SMALL ****

```

The LRECL value is obtained from the SORTIN DD statement or the WRKLRCL field in the ICHRSMFI module. See "Record reformatting" on page 122 for more details.

90.053 13:51:40		RACF REPORT - LISTING OF PROCESS RECORDS									
DATE	TIME	SYSID	*JOB/USER	*STEP/	--TERMINAL--	ID	LVL	T	L		
			NAME	GROUP							
90.053	12:15:03	R190	IBMUSER	SYS1		LE02	0	1	0	JOBID=(IBMUSER 90.053 12:15:01),USERDATA=(),OWNER=AUTH=(NONE),REASON=(NONE)	
90.053	12:15:08	R190	IBMUSER GEORGE JONES	SYS1		LE02	0	2	0	JOBID=(IBMUSER 90.053 12:15:01),USERDATA=(),OWNER=IBMUSER AUTH=(NORMAL),REASON=(ENTITY OR FAILSOFT PROCESSING) LOGSTR='LOGSTR DATA' USER SECLABEL=SYSHIGH,SESSION=TSO LOGON,TERMINAL=LE02 DATASET=SYS1.BROADCAST,GENPROF=SYS1.BROADCAST,VOLUME=SPOOL1,LEVEL=00 INTENT=READ,ALLOWED=ALTER	
90.053	12:17:33	R190	IBMUSER GEORGE JONES	SYS1		LE02	0	10	0	JOBID=(IBMUSER 90.053 12:15:01),USERDATA=(),OWNER=IBMUSER AUTH=(SPECIAL),REASON=(SPECIAL/OPERATIONS) USER SECLABEL=SYSHIGH,SESSION=TSO LOGON,TERMINAL=LE02 ADDUSER RACFU01 DFLTGRP(SYS1) PASSWORD(****) NAME('#####') AUTHORITY(USE) NOGRPACC UACC(NONE) NOADSP OWNER(IBMUSER) NOSPECIAL NOOPERATIONS NOCLAUTH NOAUDITOR NOOIDCARD	
90.053	12:17:41	R190	IBMUSER GEORGE JONES	SYS1		LE02	0	24	0	JOBID=(IBMUSER 90.053 12:15:01),USERDATA=(),OWNER=AUTH=(SPECIAL),REASON=(COMMAND) USER SECLABEL=SYSHIGH,SESSION=TSO LOGON,TERMINAL=LE02 SETROPTS STATISTICS(DATASET)	
90.053	12:17:43	R190	IBMUSER GEORGE JONES	SYS1		LE02	0	8	0	JOBID=(IBMUSER 90.053 12:15:01),USERDATA=(),OWNER=IBMUSER AUTH=(SPECIAL),REASON=(SPECIAL/OPERATIONS) USER SECLABEL=SYSHIGH,SESSION=TSO LOGON,TERMINAL=LE02 ADDSD IBMUSER.RACHECK.DATA UACC(NONE) SET NEW SECLABEL=NO SECLABEL,OLD SECLABEL=SYSHIGH	
90.053	12:17:44	R190	IBMUSER GEORGE JONES	SYS1		LE02	0	19	0	JOBID=(IBMUSER 90.053 12:15:01),USERDATA=(),OWNER=IBMUSER AUTH=(SPECIAL),REASON=(SPECIAL/OPERATIONS) USER SECLABEL=SYSHIGH,SESSION=TSO LOGON,TERMINAL=LE02 PERMIT IBMUSER.RACHECK.DATA CLASS(DATASET) ID(RACFU01) ACCESS(READ)	
90.053	12:17:49	R190	IBMUSER GEORGE JONES	SYS1		LE02	0	21	0	JOBID=(IBMUSER 90.053 12:15:01),USERDATA=(),OWNER=AUTH=(SPECIAL),REASON=(SPECIAL/OPERATIONS) USER SECLABEL=SYSHIGH,SESSION=TSO LOGON,TERMINAL=LE02 RDEFINE DIRECTRY FP.IBMUSER.DIR LEVEL(00) NONOTIFY	
90.053	12:20:09	R190	IBMUSER GEORGE JONES	SYS1		LE02	0	15	0	JOBID=(IBMUSER 90.053 12:15:01),USERDATA=(),OWNER=IBMUSER AUTH=(SPECIAL),REASON=(SPECIAL/OPERATIONS) USER SECLABEL=SYSHIGH,SESSION=TSO LOGON,TERMINAL=LE02 DELSD IBMUSER.NOACC.DATA SET NEW SECLABEL=SYSHIGH,OLD SECLABEL=NO SECLABEL	
90.053	12:26:42	R190	*IBMUSER	*RACFPROF	FFFFFF		0	1	11	JOBID=(IBMUSER 90.053 12:15:01),USERDATA=(),OWNER=AUTH=(NONE),REASON=(RACINIT FAILURE) USER SECLABEL=UNDEF,SESSION=TSO LOGON,TOKEN USER ATTRIBUTES=(UNDEFINED USER),TERMINAL=FFFFFF,SUBMITTING GROUP=GROUPA	
90.053	12:29:32	R190	IBMUSER GEORGE JONES	SYS1			0	13	0	JOBID=(IBMUSER 90.053 12:15:01),USERDATA=(),OWNER=IBMUSER AUTH=(SPECIAL),REASON=(SPECIAL/OPERATIONS) USER SECLABEL=SYSHIGH,TOKEN STATUS=(CREATED BY PRE 1.9 RACF CALL ALTUSER TS066 NOSECLABEL NEW SECLABEL=NO SECLABEL,OLD SECLABEL=L2C1	
90.053	12:29:56	R190	TS0G5 X	SYS1			0	1	1	JOBID=(IBMUSER 90.053 12:15:01),USERDATA=(),OWNER=AUTH=(NONE),REASON=(RACINIT FAILURE) USER SECLABEL=UNDEF	
90.053	12:36:49	R190	TS0G5 X	SYS1			0	1	1	JOBID=(IBMUSER 90.053 12:15:01),USERDATA=(),OWNER=AUTH=(NONE),REASON=(RACINIT FAILURE) USER SECLABEL=L1C1	
90.053	12:41:10	R190	IBMUSER	SYS1	LE02		0	1	8	JOBID=(IBMUSER 90.053 12:15:01),USERDATA=(),OWNER=AUTH=(NONE),REASON=(NONE)	
90.053	01:12:01	R190	*LISTBC	*			0	2	0	JOBID=(IBMUSER 90.053 12:15:01),USERDATA=(),OWNER=AUTH=(TRUSTED),REASON=(LOGOPTIONS) USER SECLABEL=SYSHIGH,SESSION=SYSTEM ADDRESS SPACE, TOKEN=(DEFAULT TOKEN),TOKEN USER ATTRIBUTES=(TRUSTED COMPUTER BASE) DATASET=BROADCAST.IBMUSER, VOLUME=TEMP01,LEVEL=00,INTENT=ALTER,ALLOWED=ALTER	
90.053	01:24:53	R190	IBMUSER	SYS1			0	2	0	JOBID=(IBMUSERX 90.053 12:15:01),USERDATA=(),OWNER=IBMUSER AUTH=(NORMAL),REASON=(LOGOPTIONS) SESSION=INTERNAL READER BATCH JOB,JESINPUT=INTRDR,EXENODE=N1 SUBMITTING USER=IBMUSER,SUBMITTING NODE=N1,SUBMITTING GROUP=SYS1 DATASET=SYS1.MANA,GENPROF=SYS1.*,VOLUME=PAGE08, LEVEL=00,INTENT=READ,ALLOWED=ALTER	

Figure 38. Listing of Process Records

Note: For Figure 38:

- Token-related information in the report is extracted from the Type 53 relocate sections. The format of these records is documented in *z/OS Security Server RACF Macros and Interfaces*.
- TOKEN STATUS=(CREATED BY PRE RACF 1.9 CALL) means that the TOKLT19 bit was set. This bit was set when a token is created and based on a pre-RACF 1.9 ACEE. The bit was on in the UTOKEN that was copied to the SMF record.
- The following text may appear in the report:
 **** RECORD TRUNCATED BY RACFRW – INFORMATION LOST ****

This indicates that the LRECL value on the SORTIN DD statement was too small or that the value of WRKLRCL (in the ICHRSMFI module) was too small. See "Record reformatting" on page 122 for more details.

- When a profile is not found and *BYPASS* was the user ID on RACHECK, the audit record will have the entity name, not the profile name.

```
89.196 14:23:38
```

RACF REPORT - SHORT USER SUMMARY											
		----- R E S O U R C E S T A T I S T I C S -----									
USER/ *JOB	NAME	---- JOB/LOGON ----		----- I N T E N T S -----							TOTAL
		SUCCESS	VIOLATION	SUCCESS	WARNING	VIOLATION	ALTER	CONTROL	UPDATE	READ	
*CLRMANB		1	0	0	0	0	0	0	0	0	0
IBMUSER		7	0	0	0	0	0	0	0	0	0
RACUSR1		0	0	1	0	0	0	0	0	0	1
RACUSR1	MARY BAILEY	0	0	21	0	0	21	0	0	0	21
RACUSR2		0	0	1	0	0	0	0	0	0	1
RACUSR2	MARY PURCELL	0	0	1	0	0	1	0	0	0	1
RACUSR3		0	0	1	0	0	0	0	0	0	1
RACUSR3	HARRIET BIRD	0	0	1	0	0	1	0	0	0	1
RACUSR4		0	0	1	0	0	0	0	0	0	1
RACUSR4	JOHN H. BUKOWSKI	0	0	1	0	0	1	0	0	0	1
RACUSR5		0	0	1	0	0	0	0	0	0	1
RACUSR5	MELANIE WILKES	0	0	1	0	0	1	0	0	0	1
RACUSR6		0	0	1	0	0	0	0	0	0	1
RACUSR6	FRED PRETOCK	0	0	1	0	0	1	0	0	0	1
RACUSR7		0	0	1	0	0	0	0	0	0	1
RACUSR7	HESTER WILSON	0	0	1	0	0	1	0	0	0	1
SLCUSRD1		0	0	1	0	0	0	0	0	1	1
SLCUSRD5		0	0	0	0	1	0	0	0	1	1
ACCUMULATED TOTALS -		8	0	35	0	1	27	0	0	2	36
PERCENTAGE OF TOTAL ACCESSES -				97 %	0 %	3 %	75 %	0 %	0 %	6 %	
UNDEFINED USERS (JOBS) ONLY											
ACCUMULATED TOTALS -		1	0	0	0	0	0	0	0	0	0
PERCENTAGE OF TOTAL ACCESSES -				0 %	0 %	0 %	0 %	0 %	0 %	0 %	

Figure 39. Short User Summary Report

```
89.196 14:23:38
```

RACF REPORT - SHORT GROUP SUMMARY											
		----- R E S O U R C E S T A T I S T I C S -----									
GROUP/ *STEP		---- JOB/LOGON ----		----- I N T E N T S -----							TOTAL
		SUCCESS	VIOLATION	SUCCESS	WARNING	VIOLATION	ALTER	CONTROL	UPDATE	READ	
**		1	0	0	0	0	0	0	0	0	0
SYS1		7	0	35	0	1	27	0	0	2	36
ACCUMULATED TOTALS -		8	0	35	0	1	27	0	0	2	36
PERCENTAGE OF TOTAL ACCESSES -				97 %	0 %	3 %	75 %	0 %	0 %	6 %	
UNDEFINED USERS (JOBS) ONLY											
ACCUMULATED TOTALS -		1	0	0	0	0	0	0	0	0	0
PERCENTAGE OF TOTAL ACCESSES -				0 %	0 %	0 %	0 %	0 %	0 %	0 %	

Figure 40. Short Group Summary Report

RESOURCE NAME	SUCCESS	WARNING	VIOLATION	I N T E N T S				TOTAL
				ALTER	CONTROL	UPDATE	READ	
CLASS = DATASET								
RACUSR1.NEW.DS1	3	0	0	2	0	0	0	3
RACUSR2.NEW.DS2	3	0	0	2	0	0	0	3
RACUSR3.NEW.DS3	3	0	0	2	0	0	0	3
RACUSR4.NEW.DS4	3	0	0	2	0	0	0	3
RACUSR5.NEW.DS5	3	0	0	2	0	0	0	3
RACUSR6.NEW.DS6	3	0	0	2	0	0	0	3
RACUSR7.NEW.DS7	3	0	0	2	0	0	0	3
SLCUSRD0.SLCDSND0	2	0	0	2	0	0	0	2
SLCUSRD1.SLCDSND1	3	0	0	2	0	0	1	3
SLCUSRD3.SLCDSND3	2	0	0	2	0	0	0	2
SLCUSRD4.SLCDSND4	2	0	0	2	0	0	0	2
SLCUSRD5.SLCDSND5	2	0	1	2	0	0	1	3
CLASS = SECDATA								
SECLEVEL	3	0	0	3	0	0	0	3
ACCUMULATED TOTALS -	35	0	1	27	0	0	2	36
PERCENTAGE OF TOTAL ACCESSES -	97 %	0 %	3 %	75 %	0 %	0 %	6 %	
GENERIC PROFILE USED								
ACCUMULATED TOTALS -	0	0	0	0	0	0	0	0
PERCENTAGE OF TOTAL ACCESSES -	0 %	0 %	0 %	0 %	0 %	0 %	0 %	

Figure 41. Short Resource Summary Report

Note: In this example, the SUCCESS number is shown as 3, while the ALTER column shows 2. This occurred because the report writer also processed other events – in this case DEFINE events. Keep in mind the totals reflect the number of records processed, not just the number listed. Therefore, there may be data recorded in the totals that are not listed specifically in the report itself.

90.053 13:51:40

RACF REPORT - SHORT COMMAND SUMMARY
OCCURRENCES

QUALIFIER	
EVENT = 8 - ADDSD COMMAND	
0 - NO VIOLATIONS DETECTED	3
ACCUMULATED TOTALS -	3
EVENT = 9 - ADDGROUP COMMAND	
0 - NO VIOLATIONS DETECTED	1
ACCUMULATED TOTALS -	1
EVENT = 10 - ADDUSER COMMAND	
0 - NO VIOLATIONS DETECTED	7
ACCUMULATED TOTALS -	7
EVENT = 13 - ALTUSER COMMAND	
0 - NO VIOLATIONS DETECTED	1
ACCUMULATED TOTALS -	1
EVENT = 14 - CONNECT COMMAND	
0 - NO VIOLATIONS DETECTED	1
ACCUMULATED TOTALS	1
EVENT = 15 - DELDSD COMMAND	
0 - NO VIOLATIONS DETECTED	2
ACCUMULATED TOTALS -	2
EVENT = 16 - DELGROUP COMMAND	
0 - NO VIOLATIONS DETECTED	1
ACCUMULATED TOTALS -	1
EVENT = 17 - DELUSER COMMAND	
0 - NO VIOLATIONS DETECTED	1
ACCUMULATED TOTALS -	1
EVENT = 19 - PERMIT COMMAND	
0 - NO VIOLATIONS DETECTED	14
ACCUMULATED TOTALS	14
EVENT = 20 - RALTER COMMAND	
0 - NO VIOLATIONS DETECTED	2
ACCUMULATED TOTALS -	2
EVENT = 21 - RDEFINE COMMAND	
0 - NO VIOLATIONS DETECTED	7
ACCUMULATED TOTALS -	7
EVENT = 22 - RDELETE COMMAND	
0 - NO VIOLATIONS DETECTED	7
ACCUMULATED TOTALS -	7
EVENT = 23 - REMOVE COMMAND	
0 - NO VIOLATIONS DETECTED	1
ACCUMULATED TOTALS	1
EVENT = 24 - SETROPTS COMMAND	
0 - NO VIOLATIONS DETECTED	49
ACCUMULATED TOTALS -	49

Figure 42. Short Command Summary Report

90.053 13:51:40

RACF REPORT - SHORT EVENT SUMMARY
OCCURRENCES

EVENT =	QUALIFIER	OCCURRENCES
1	JOB INITIATION / TSO LOGON	
	0 - SUCCESSFUL INITIATION/LOGON	4
	1 - INVALID PASSWORD	1
	8 - SUCCESSFUL TERMINATION	5
	10-INSUFF. SECURITY LABEL AUTHORITY	4
	11-NOT AUTHORIZED TO SECURITY LABEL	3
	18-SUBMITTER UNAUTHOR. TO SEC. LABEL	1
	26-INVALID NEW PASSWORD	1
	ACCUMULATED TOTALS -	19
2	RESOURCE ACCESS	
	0 - SUCCESSFUL ACCESS	9
	1 - INSUFFICIENT AUTHORITY	1
	ACCUMULATED TOTALS -	10
8	ADSD COMMAND	
	0 - NO VIOLATIONS DETECTED	3
	ACCUMULATED TOTALS -	3
9	ADDGROUP COMMAND	
	0 - NO VIOLATIONS DETECTED	1
	ACCUMULATED TOTALS -	1
10	ADDUSER COMMAND	
	0 - NO VIOLATIONS DETECTED	7
	ACCUMULATED TOTALS -	7
13	ALTUSER COMMAND	
	0 - NO VIOLATIONS DETECTED	1
	ACCUMULATED TOTALS -	1
14	CONNECT COMMAND	
	0 - NO VIOLATIONS DETECTED	1
	ACCUMULATED TOTALS -	1
15	DELDSD COMMAND	
	0 - NO VIOLATIONS DETECTED	2
	ACCUMULATED TOTALS -	2
16	DELGROUP COMMAND	
	0 - NO VIOLATIONS DETECTED	1
	ACCUMULATED TOTALS	1
17	DELUSER COMMAND	
	0 - NO VIOLATIONS DETECTED	1
	ACCUMULATED TOTALS -	1
19	PERMIT COMMAND	
	0 - NO VIOLATIONS DETECTED	14
	ACCUMULATED TOTALS -	14
20	RALTER COMMAND	
	0 - NO VIOLATIONS DETECTED	2
	ACCUMULATED TOTALS -	2
21	RDEFINE COMMAND	
	0 - NO VIOLATIONS DETECTED	7
	ACCUMULATED TOTALS -	7
22	RDELETE COMMAND	
	0 - NO VIOLATIONS DETECTED	7
	ACCUMULATED TOTALS -	7
23	REMOVE COMMAND	
	0 - NO VIOLATIONS DETECTED	1
	ACCUMULATED TOTALS -	1
24	SETROPTS COMMAND	
	0 - NO VIOLATIONS DETECTED	49
	ACCUMULATED TOTALS -	49
	ACCUMULATED TOTALS -	126

Figure 43. Short Event Summary Report

89.196 14:23:38

RACF REPORT - SHORT OWNER SUMMARY

OWNER	----- I N T E N T S -----							TOTAL
	SUCCESS	WARNING	VIOLATION	ALTER	CONTROL	UPDATE	READ	
RACUSR1	6	0	0	5	0	0	0	6
RACUSR2	3	0	0	2	0	0	0	3
RACUSR3	3	0	0	2	0	0	0	3
RACUSR4	3	0	0	2	0	0	0	3
RACUSR5	3	0	0	2	0	0	0	3
RACUSR6	3	0	0	2	0	0	0	3
RACUSR7	3	0	0	2	0	0	0	3
SLCUSRD0	2	0	0	2	0	0	0	2
SLCUSRD1	3	0	0	2	0	0	1	3
SLCUSRD3	2	0	0	2	0	0	0	2
SLCUSRD4	2	0	0	2	0	0	0	2
SLCUSRD5	2	0	1	2	0	0	1	3
ACCUMULATED TOTALS -	35	0	1	27	0	0	2	36
PERCENTAGE OF TOTAL ACCESSES -	97 %	0 %	3 %	75 %	0 %	0 %	6 %	

Figure 44. Short Owner Summary Report

89.218 12:36:12

RACF REPORT - USER BY RESOURCE SUMMARY

RESOURCE NAME	----- I N T E N T S -----							TOTAL
	SUCCESS	WARNING	VIOLATION	ALTER	CONTROL	UPDATE	READ	
USER = IBMUSER								
NAME = JOHN P. ZILLER								
CLASS = SECADATA								
SECLEVEL	1	0	0	1	0	0	0	1
ACCUMULATED TOTALS -	1	0	0	1	0	0	0	1
PERCENTAGE OF TOTAL ACCESSES -	100 %	0 %	0 %	100 %	0 %	0 %	0 %	
GENERIC PROFILE USED								
ACCUMULATED TOTALS -	0	0	0	0	0	0	0	0
PERCENTAGE OF TOTAL ACCESSES -	0 %	0 %	0 %	0 %	0 %	0 %	0 %	
USER = RACUSR1								
CLASS = DATASET								
RACUSR1.NEW.DS1	0	0	0	0	0	0	0	0
NAME = MARY BAILEY								
CLASS = DATASET								
RACUSR1.NEW.DS1	2	0	0	1	0	0	0	2
RACUSR1.SMFS23	2	0	0	2	0	0	0	2
NAME = MARY BAILEY								
CLASS = SECADATA								
SECLEVEL	5	0	0	5	0	0	0	5
ACCUMULATED TOTALS -	9	0	0	8	0	0	0	9
PERCENTAGE OF TOTAL ACCESSES -	100 %	0 %	0 %	89 %	0 %	0 %	0 %	
GENERIC PROFILE USED								
ACCUMULATED TOTALS -	0	0	0	0	0	0	0	0
PERCENTAGE OF TOTAL ACCESSES -	0 %	0 %	0 %	0 %	0 %	0 %	0 %	
USER = RACUSR2								
CLASS = DATASET								
RACUSR2.NEW.DS2	0	0	0	0	0	0	0	0
NAME = JOHN P. ZILLER								
CLASS = DATASET								
RACUSR2.NEW.DS2	2	0	0	1	0	0	0	2
ACCUMULATED TOTALS -	2	0	0	1	0	0	0	2
PERCENTAGE OF TOTAL ACCESSES -	100 %	0 %	0 %	50 %	0 %	0 %	0 %	
GENERIC PROFILE USED								
ACCUMULATED TOTALS -	0	0	0	0	0	0	0	0
PERCENTAGE OF TOTAL ACCESSES -	0 %	0 %	0 %	0 %	0 %	0 %	0 %	
USER = RACUSR3								
CLASS = DATASET								

Figure 45. User by Resource Summary Report

89.218 12:36:12

RACF REPORT - GROUP BY RESOURCE SUMMARY

RESOURCE NAME	SUCCESS	WARNING	VIOLATION	I N T E N T S				TOTAL
				ALTER	CONTROL	UPDATE	READ	
GROUP = SYS1								
CLASS = DATASET								
RACUSR1.NEW.DS1	2	0	0	1	0	0	0	2
RACUSR1.SMFS23	2	0	0	2	0	0	0	2
RACUSR2.NEW.DS2	2	0	0	1	0	0	0	2
RACUSR3.NEW.DS3	2	0	0	1	0	0	0	2
RACUSR4.NEW.DS4	2	0	0	1	0	0	0	2
RACUSR5.NEW.DS5	2	0	0	1	0	0	0	2
RACUSR6.NEW.DS6	2	0	0	1	0	0	0	2
SLCUSRD1.SLCDSND1	2	0	0	0	0	0	2	2
SLCUSRD3.SLCDSND3	1	0	0	0	0	0	1	1
SLCUSRD5.SLCDSND5	0	0	2	0	0	0	2	2
CLASS = SECADATA								
SECLEVEL	6	0	0	6	0	0	0	6
ACCUMULATED TOTALS -	23	0	2	14	0	0	5	25
PERCENTAGE OF TOTAL ACCESSES -	92 %	0 %	8 %	56 %	0 %	0 %	20 %	
GENERIC PROFILE USED								
ACCUMULATED TOTALS -	0	0	0	0	0	0	0	0
PERCENTAGE OF TOTAL ACCESSES -	0 %	0 %	0 %	0 %	0 %	0 %	0 %	

Figure 46. Group by Resource Summary Report

89.218 12:36:12

RACF REPORT - RESOURCE BY USER SUMMARY

USER/ *JOB	SUCCESS	WARNING	VIOLATION	I N T E N T S				TOTAL
				ALTER	CONTROL	UPDATE	READ	
DATASET = RACUSR1.NEW.DS1								
RACUSR1 MARY BAILEY	2	0	0	1	0	0	0	2
ACCUMULATED TOTALS -	2	0	0	1	0	0	0	2
PERCENTAGE OF TOTAL ACCESSES -	100 %	0 %	0 %	50 %	0 %	0 %	0 %	
UNDEFINED USERS (JOBS) ONLY								
ACCUMULATED TOTALS -	0	0	0	0	0	0	0	0
PERCENTAGE OF TOTAL ACCESSES -	0 %	0 %	0 %	0 %	0 %	0 %	0 %	
GENERIC PROFILE USED								
ACCUMULATED TOTALS -	0	0	0	0	0	0	0	0
PERCENTAGE OF TOTAL ACCESSES -	0 %	0 %	0 %	0 %	0 %	0 %	0 %	
DATASET = RACUSR1.SMFS23								
RACUSR1 MARY BAILEY	2	0	0	2	0	0	0	2
ACCUMULATED TOTALS -	2	0	0	2	0	0	0	2
PERCENTAGE OF TOTAL ACCESSES -	100 %	0 %	0 %	100 %	0 %	0 %	0 %	
UNDEFINED USERS (JOBS) ONLY								
ACCUMULATED TOTALS -	0	0	0	0	0	0	0	0
PERCENTAGE OF TOTAL ACCESSES -	0 %	0 %	0 %	0 %	0 %	0 %	0 %	
GENERIC PROFILE USED								
ACCUMULATED TOTALS -	0	0	0	0	0	0	0	0
PERCENTAGE OF TOTAL ACCESSES -	0 %	0 %	0 %	0 %	0 %	0 %	0 %	
DATASET = RACUSR2.NEW.DS2								
RACUSR2 JOHN P. ZILLER	2	0	0	1	0	0	0	2
ACCUMULATED TOTALS -	2	0	0	1	0	0	0	2
PERCENTAGE OF TOTAL ACCESSES -	100 %	0 %	0 %	50 %	0 %	0 %	0 %	
UNDEFINED USERS (JOBS) ONLY								
ACCUMULATED TOTALS -	0	0	0	0	0	0	0	0
PERCENTAGE OF TOTAL ACCESSES -	0 %	0 %	0 %	0 %	0 %	0 %	0 %	
GENERIC PROFILE USED								
ACCUMULATED TOTALS -	0	0	0	0	0	0	0	0
PERCENTAGE OF TOTAL ACCESSES -	0 %	0 %	0 %	0 %	0 %	0 %	0 %	
DATASET = RACUSR3.NEW.DS3								
RACUSR3 HARRIET BIRD	2	0	0	1	0	0	0	2
ACCUMULATED TOTALS -	2	0	0	1	0	0	0	2
PERCENTAGE OF TOTAL ACCESSES -	100 %	0 %	0 %	50 %	0 %	0 %	0 %	
UNDEFINED USERS (JOBS) ONLY								
ACCUMULATED TOTALS -	0	0	0	0	0	0	0	0
PERCENTAGE OF TOTAL ACCESSES -	0 %	0 %	0 %	0 %	0 %	0 %	0 %	
GENERIC PROFILE USED								
ACCUMULATED TOTALS -	0	0	0	0	0	0	0	0
PERCENTAGE OF TOTAL ACCESSES -	0 %	0 %	0 %	0 %	0 %	0 %	0 %	
DATASET = RACUSR4.NEW.DS4								
RACUSR4 JOHN H. BUKOWSKI	2	0	0	1	0	0	0	2
ACCUMULATED TOTALS -	2	0	0	1	0	0	0	2
PERCENTAGE OF TOTAL ACCESSES -	100 %	0 %	0 %	50 %	0 %	0 %	0 %	
UNDEFINED USERS (JOBS) ONLY								
ACCUMULATED TOTALS -	0	0	0	0	0	0	0	0

Figure 47. Resource by User Summary Report

89.218 12:36:12

RACF REPORT - RESOURCE BY GROUP SUMMARY

GROUP/ *STEP	----- I N T E N T S -----							TOTAL
	SUCCESS	WARNING	VIOLATION	ALTER	CONTROL	UPDATE	READ	
DATASET = RACUSR1.NEW.DS1								
SYS1	2	0	0	1	0	0	0	2
ACCUMULATED TOTALS -	2	0	0	1	0	0	0	2
PERCENTAGE OF TOTAL ACCESSES -	100 %	0 %	0 %	50 %	0 %	0 %	0 %	
UNDEFINED USERS (JOBS) ONLY								
ACCUMULATED TOTALS -	0	0	0	0	0	0	0	0
PERCENTAGE OF TOTAL ACCESSES -	0 %	0 %	0 %	0 %	0 %	0 %	0 %	
GENERIC PROFILE USED								
ACCUMULATED TOTALS -	0	0	0	0	0	0	0	0
PERCENTAGE OF TOTAL ACCESSES -	0 %	0 %	0 %	0 %	0 %	0 %	0 %	
DATASET = RACUSR1.SMFS23								
SYS1	2	0	0	2	0	0	0	2
ACCUMULATED TOTALS -	2	0	0	2	0	0	0	2
PERCENTAGE OF TOTAL ACCESSES -	100 %	0 %	0 %	100 %	0 %	0 %	0 %	
UNDEFINED USERS (JOBS) ONLY								
ACCUMULATED TOTALS -	0	0	0	0	0	0	0	0
PERCENTAGE OF TOTAL ACCESSES -	0 %	0 %	0 %	0 %	0 %	0 %	0 %	
GENERIC PROFILE USED								
ACCUMULATED TOTALS -	0	0	0	0	0	0	0	0
PERCENTAGE OF TOTAL ACCESSES -	0 %	0 %	0 %	0 %	0 %	0 %	0 %	
DATASET = RACUSR2.NEW.DS2								
SYS1	2	0	0	1	0	0	0	2
ACCUMULATED TOTALS -	2	0	0	1	0	0	0	2
PERCENTAGE OF TOTAL ACCESSES -	100 %	0 %	0 %	50 %	0 %	0 %	0 %	
UNDEFINED USERS (JOBS) ONLY								
ACCUMULATED TOTALS -	0	0	0	0	0	0	0	0
PERCENTAGE OF TOTAL ACCESSES -	0 %	0 %	0 %	0 %	0 %	0 %	0 %	
GENERIC PROFILE USED								
ACCUMULATED TOTALS -	0	0	0	0	0	0	0	0
PERCENTAGE OF TOTAL ACCESSES -	0 %	0 %	0 %	0 %	0 %	0 %	0 %	
DATASET = RACUSR3.NEW.DS3								
SYS1	2	0	0	1	0	0	0	2
ACCUMULATED TOTALS -	2	0	0	1	0	0	0	2
PERCENTAGE OF TOTAL ACCESSES -	100 %	0 %	0 %	50 %	0 %	0 %	0 %	
UNDEFINED USERS (JOBS) ONLY								
ACCUMULATED TOTALS -	0	0	0	0	0	0	0	0
PERCENTAGE OF TOTAL ACCESSES -	0 %	0 %	0 %	0 %	0 %	0 %	0 %	
GENERIC PROFILE USED								
ACCUMULATED TOTALS -	0	0	0	0	0	0	0	0
PERCENTAGE OF TOTAL ACCESSES -	0 %	0 %	0 %	0 %	0 %	0 %	0 %	
DATASET = RACUSR4.NEW.DS4								
SYS1	2	0	0	1	0	0	0	2
ACCUMULATED TOTALS -	2	0	0	1	0	0	0	2
PERCENTAGE OF TOTAL ACCESSES -	100 %	0 %	0 %	50 %	0 %	0 %	0 %	
UNDEFINED USERS (JOBS) ONLY								
ACCUMULATED TOTALS -	0	0	0	0	0	0	0	0

Figure 48. Resource by Group Summary Report

89.218 12:36:12

RACF REPORT - RESOURCE BY EVENT SUMMARY
OCCURRENCES

EVENT/QUALIFIER	OCCURRENCES
DATASET = RACUSR1.NEW.DS1	
7 - DEFINE RESOURCE	
0 - SUCCESSFUL DEFINITION	1
ACCUMULATED TOTALS -	1
GENERIC PROFILE USED	
ACCUMULATED TOTALS -	0
8 - ADDSD COMMAND	
0 - NO VIOLATIONS DETECTED	1
ACCUMULATED TOTALS -	1
GENERIC PROFILE USED	
ACCUMULATED TOTALS -	0
ACCUMULATED TOTALS -	2
GENERIC PROFILE USED	
ACCUMULATED TOTALS -	0
DATASET = RACUSR1.SMFS23	
8 - ADDSD COMMAND	
0 - NO VIOLATIONS DETECTED	1
ACCUMULATED TOTALS -	1
GENERIC PROFILE USED	
ACCUMULATED TOTALS -	0
19 - PERMIT COMMAND	
0 - NO VIOLATIONS DETECTED	1
ACCUMULATED TOTALS -	1
GENERIC PROFILE USED	
ACCUMULATED TOTALS -	0
ACCUMULATED TOTALS -	2
GENERIC PROFILE USED	
ACCUMULATED TOTALS -	0
DATASET = RACUSR2.NEW.DS2	
7 - DEFINE RESOURCE	
0 - SUCCESSFUL DEFINITION	1
ACCUMULATED TOTALS -	1
GENERIC PROFILE USED	
ACCUMULATED TOTALS -	0
8 - ADDSD COMMAND	
0 - NO VIOLATIONS DETECTED	1
ACCUMULATED TOTALS -	1
GENERIC PROFILE USED	
ACCUMULATED TOTALS -	0

Figure 49. Resource by Event Summary Report

```

89.218 12:36:12
                                QUALIFIER
EVENT = 2 - RESOURCE ACCESS
      0 - SUCCESSFUL ACCESS
                                OCCURRENCES
                                RESOURCE
                                DATASET = SLCUSRD1.SLCDSND1
                                DATASET = SLCUSRD3.SLCDSND3
                                ACCUMULATED TOTALS -
                                GENERIC PROFILE USED
                                ACCUMULATED TOTALS -
                                UNKNOWN EVENT CODE QUALIFIER
                                DATASET = SLCUSRD5.SLCDSND5
                                ACCUMULATED TOTALS -
                                GENERIC PROFILE USED
                                ACCUMULATED TOTALS -
                                ACCUMULATED TOTALS -
                                GENERIC PROFILE USED
                                ACCUMULATED TOTALS -
EVENT = 7 - DEFINE RESOURCE
      0 - SUCCESSFUL DEFINITION
                                DATASET = RACUSR1.NEW.DS1
                                DATASET = RACUSR2.NEW.DS2
                                DATASET = RACUSR3.NEW.DS3
                                DATASET = RACUSR4.NEW.DS4
                                DATASET = RACUSR5.NEW.DS5
                                DATASET = RACUSR6.NEW.DS6
                                ACCUMULATED TOTALS -
                                GENERIC PROFILE USED
                                ACCUMULATED TOTALS -
                                ACCUMULATED TOTALS -
                                GENERIC PROFILE USED
                                ACCUMULATED TOTALS -
EVENT = 8 - ADDSD COMMAND
      0 - NO VIOLATIONS DETECTED
                                DATASET = RACUSR1.NEW.DS1
                                DATASET = RACUSR1.SMFS23
                                DATASET = RACUSR2.NEW.DS2
                                DATASET = RACUSR3.NEW.DS3
                                DATASET = RACUSR4.NEW.DS4
                                DATASET = RACUSR5.NEW.DS5
                                DATASET = RACUSR6.NEW.DS6
                                ACCUMULATED TOTALS -
                                GENERIC PROFILE USED
                                ACCUMULATED TOTALS -
                                ACCUMULATED TOTALS -
                                GENERIC PROFILE USED

```

Figure 50. Event by Resource Summary Report

89.218 12:36:12

QUALIFIER	RACF REPORT - COMMAND BY USER SUMMARY
	OCCURRENCES USER NAME
EVENT = 8 - ADDSD COMMAND	
0 - NO VIOLATIONS DETECTED	
	2 RACUSR1 MARY BAILEY
	1 RACUSR2 JOHN P. ZILLER
	1 RACUSR3 HARRIET BIRD
	1 RACUSR4 JOHN H. BUKOWSKI
	1 RACUSR5 MELANIE WILKES
	1 RACUSR6
ACCUMULATED TOTALS -	7
ACCUMULATED TOTALS -	7
EVENT = 10 - ADDUSER COMMAND	
0 - NO VIOLATIONS DETECTED	
	0 IBMUSER
	1 IBMUSER THOR
	6 RACUSR1 MARY BAILEY
ACCUMULATED TOTALS -	7
1 - INSUFFICIENT AUTHORITY	
	1 RACUSR7 HESTER WILSON
	2 SLCUSRD0 (NAME UNKNOWN)
	2 SLCUSRD1 (NAME UNKNOWN)
	2 SLCUSRD3 (NAME UNKNOWN)
	2 SLCUSRD4 (NAME UNKNOWN)
	2 SLCUSRD5 (NAME UNKNOWN)
	2 SLCUSRD6 (NAME UNKNOWN)
ACCUMULATED TOTALS -	13
ACCUMULATED TOTALS -	20
EVENT = 13 - ALTUSER COMMAND	
0 - NO VIOLATIONS DETECTED	
	0 IBMUSER
	1 IBMUSER THOR
	21 RACUSR1 MARY BAILEY
ACCUMULATED TOTALS -	22
ACCUMULATED TOTALS -	22
EVENT = 17 - DELUSER COMMAND	
0 - NO VIOLATIONS DETECTED	
	0 IBMUSER
	1 IBMUSER THOR
ACCUMULATED TOTALS -	1
ACCUMULATED TOTALS -	1

Figure 51. Command by User Summary Report

```

89.218 12:36:12
                                QUALIFIER
EVENT = 8 - ADDSD COMMAND
          0 - NO VIOLATIONS DETECTED
                                7
                                SYS1
          ACCUMULATED TOTALS -
          ACCUMULATED TOTALS -
          7
EVENT = 10 - ADDUSER COMMAND
          0 - NO VIOLATIONS DETECTED
                                7
                                SYS1
          ACCUMULATED TOTALS -
          1 - INSUFFICIENT AUTHORITY
          7
          ACCUMULATED TOTALS -
          ACCUMULATED TOTALS -
          13
          13
EVENT = 13 - ALTUSER COMMAND
          0 - NO VIOLATIONS DETECTED
                                20
                                SYS1
          ACCUMULATED TOTALS -
          ACCUMULATED TOTALS -
          22
          22
EVENT = 17 - DELUSER COMMAND
          0 - NO VIOLATIONS DETECTED
                                1
                                SYS1
          ACCUMULATED TOTALS -
          ACCUMULATED TOTALS -
          1
          1
EVENT = 19 - PERMIT COMMAND
          0 - NO VIOLATIONS DETECTED
                                1
                                SYS1
          ACCUMULATED TOTALS -
          ACCUMULATED TOTALS -
          1
          1
EVENT = 20 - RALTER COMMAND
          0 - NO VIOLATIONS DETECTED
                                1
                                SYS1
          ACCUMULATED TOTALS -
          1

```

Figure 52. Command by Group Summary Report

89.218 12:36:12

RACF REPORT - COMMAND BY RESOURCE SUMMARY
OCCURRENCES RESOURCE

QUALIFIER	OCCURRENCES	RESOURCE
EVENT = 8 - ADDSD COMMAND		
0 - NO VIOLATIONS DETECTED		
	1	DATASET = RACUSR1.NEW.DS1
	1	DATASET = RACUSR1.SMFS23
	1	DATASET = RACUSR2.NEW.DS2
	1	DATASET = RACUSR3.NEW.DS3
	1	DATASET = RACUSR4.NEW.DS4
	1	DATASET = RACUSR5.NEW.DS5
	1	DATASET = RACUSR6.NEW.DS6
ACCUMULATED TOTALS - GENERIC PROFILE USED	7	
ACCUMULATED TOTALS -	0	
ACCUMULATED TOTALS - GENERIC PROFILE USED	7	
ACCUMULATED TOTALS -	0	
EVENT = 19 - PERMIT COMMAND		
0 - NO VIOLATIONS DETECTED		
	1	DATASET = RACUSR1.SMFS23
ACCUMULATED TOTALS - GENERIC PROFILE USED	1	
ACCUMULATED TOTALS -	0	
ACCUMULATED TOTALS - GENERIC PROFILE USED	1	
ACCUMULATED TOTALS -	0	
EVENT = 20 - RALTER COMMAND		
0 - NO VIOLATIONS DETECTED		
	1	SECDATA = SECLEVEL
ACCUMULATED TOTALS - GENERIC PROFILE USED	1	
ACCUMULATED TOTALS -	0	
ACCUMULATED TOTALS - GENERIC PROFILE USED	1	
ACCUMULATED TOTALS -	0	
EVENT = 21 - RDEFINE COMMAND		
0 - NO VIOLATIONS DETECTED		
	3	SECDATA = SECLEVEL
ACCUMULATED TOTALS - GENERIC PROFILE USED	3	
ACCUMULATED TOTALS -	0	
ACCUMULATED TOTALS - GENERIC PROFILE USED	3	

Figure 53. Command by Resource Summary Report

RACF REPORT - OWNER BY RESOURCE SUMMARY

RESOURCE NAME	I N T E N T S							TOTAL
	SUCCESS	WARNING	VIOLATION	ALTER	CONTROL	UPDATE	READ	
OWNER = IBMUSER								
CLASS = SECADATA								
SECLEVEL	1	0	0	1	0	0	0	1
ACCUMULATED TOTALS -	1	0	0	1	0	0	0	1
PERCENTAGE OF TOTAL ACCESSES -	100 %	0 %	0 %	100 %	0 %	0 %	0 %	
GENERIC PROFILE USED								
ACCUMULATED TOTALS -	0	0	0	0	0	0	0	0
PERCENTAGE OF TOTAL ACCESSES -	0 %	0 %	0 %	0 %	0 %	0 %	0 %	
OWNER = RACUSR1								
CLASS = DATASET								
RACUSR1.NEW.DS1	2	0	0	1	0	0	0	2
RACUSR1.SMFS23	2	0	0	2	0	0	0	2
CLASS = SECADATA								
SECLEVEL	5	0	0	5	0	0	0	5
ACCUMULATED TOTALS -	9	0	0	8	0	0	0	9
PERCENTAGE OF TOTAL ACCESSES -	100 %	0 %	0 %	89 %	0 %	0 %	0 %	
GENERIC PROFILE USED								
ACCUMULATED TOTALS -	0	0	0	0	0	0	0	0
PERCENTAGE OF TOTAL ACCESSES -	0 %	0 %	0 %	0 %	0 %	0 %	0 %	
OWNER = RACUSR2								
CLASS = DATASET								
RACUSR2.NEW.DS2	2	0	0	1	0	0	0	2
ACCUMULATED TOTALS -	2	0	0	1	0	0	0	2
PERCENTAGE OF TOTAL ACCESSES -	100 %	0 %	0 %	50 %	0 %	0 %	0 %	
GENERIC PROFILE USED								
ACCUMULATED TOTALS -	0	0	0	0	0	0	0	0
PERCENTAGE OF TOTAL ACCESSES -	0 %	0 %	0 %	0 %	0 %	0 %	0 %	
OWNER = RACUSR3								
CLASS = DATASET								
RACUSR3.NEW.DS3	2	0	0	1	0	0	0	2
ACCUMULATED TOTALS -	2	0	0	1	0	0	0	2
PERCENTAGE OF TOTAL ACCESSES -	100 %	0 %	0 %	50 %	0 %	0 %	0 %	
GENERIC PROFILE USED								
ACCUMULATED TOTALS -	0	0	0	0	0	0	0	0
PERCENTAGE OF TOTAL ACCESSES -	0 %	0 %	0 %	0 %	0 %	0 %	0 %	
OWNER = RACUSR4								
CLASS = DATASET								
RACUSR4.NEW.DS4	2	0	0	1	0	0	0	2
ACCUMULATED TOTALS -	2	0	0	1	0	0	0	2
PERCENTAGE OF TOTAL ACCESSES -	100 %	0 %	0 %	50 %	0 %	0 %	0 %	

Figure 54. Owner by Resource Summary Report

Merging SMF records produced by RACF for z/VM with SMF records produced by RACF for MVS

Although the content of RACF's audit records is the same in z/VM as it is in MVS, the record format is slightly different. Therefore, if you want to merge the SMF records produced by RACF for z/VM with those produced by RACF for MVS, the z/VM records must be reformatted. For more information, see the specific RACF Auditor's Guide you use with your z/VM system.

Appendix B. XML Schema

The following is the XML Schema document for RACF, IRRSCHEM.XSD. Since this can change from time to time, you should look at IRRSCHEM in SYS1.SAMPLIB to find the current level of this information.

```
<?xml version="1.0" encoding="ebcdic-cp-us" ?>
<xs:schema targetNamespace="http://www.ibm.com/xmlns/zOS/IRRSchema"
  elementFormDefault="qualified" attributeFormDefault="unqualified"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="http://www.ibm.com/xmlns/zOS/IRRSchema">

  <!-- ----- -->
  <!-- -->
  <!-- PROPRIETARY STATEMENT -->
  <!-- -->
  <!-- -->
  <!-- -->
  <!-- Licensed Materials - Property of IBM -->
  <!-- 5650-ZOS Copyright IBM Corp. 2005, 2009 -->
  <!-- -->
  <!-- STATUS= HRF7760 -->
  <!-- -->
  <!-- END_OF_PROPRIETARY_STATEMENT -->
  <!-- -->
  <!-- ----- -->
  <!-- -->
  <!--*01* EXTERNAL CLASSIFICATION: OTHER -->
  <!--*01* END OF EXTERNAL CLASSIFICATION: -->
  <!-- -->
  <!-- This SAMPLIB member is only an example. The value -->
  <!-- on each statement is not necessarily an IBM-recommended -->
  <!-- value. Installations may use this member to validate XML -->
  <!-- documents produced by the RACF SMF Data Unload Utility -->
  <!-- -->
  <!-- ----- -->
  <!-- -->
  <!-- Name: IRRSCHEM -->
  <!-- -->
  <!-- Description: Define the XML grammar used to validate the XML -->
  <!-- instance documents produced by the RACF SMF Data -->
  <!-- Unload Utility (IRRADU00). -->
  <!-- -->
  <!-- Operation: This is an XML schema document that defines the -->
  <!-- XML tag language used by the RACF SMF Data -->
  <!-- Unload Utility(IRRADU00) XML instance documents. -->
  <!-- It pulls in the schema definitions for the -->
  <!-- meta data and EIM specific elements. -->
  <!-- -->
  <!-- Notes: -->
  <!-- -->
  <!-- The XML tag names are derived from the DB2 field names -->
  <!-- using the following rules: -->
  <!-- -->
  <!-- - Remove the event prefix, keeping the characters following -->
  <!-- the first underscore. -->
  <!-- - Fold all letters to lowercase, except the first character -->
  <!-- after each remaining underscore. -->
  <!-- - Remove the remaining underscores, compressing the string. -->
  <!-- -->
  <!-- Example: -->
  <!-- ACC_EVENT_TYPE eventType -->
  <!-- -->
  <!-- -->
```

```

<!-- The following tags are exceptions: ---
<!-- ---
<!-- DB2 Field XML Tag ---
<!-- - - - - - ---
<!-- RINI_TERM riniTerm ---
<!-- CAUD_REQUEST_READ caudRequestRead ---
<!-- CAUD_REQUEST_WRITE caudRequestWrite ---
<!-- CAUD_REQUEST_EXEC caudRequestExec ---
<!-- SSCL_OLDSECL oldSec1 ---
<!-- KTKT_PRINCIPAL kerbPrincipal ---
<!-- PDAC_PRINCIPAL pdasPrincipal ---
<!-- ACC_NAME profileName ---
<!-- APPC_NAME profileName ---
<!-- *_SECL_LINK eventLink ---
<!-- *_LOGSTRING logstr ---
<!-- *RESERVED* ** no tag ** ---
<!-- ---
<!-- CHANGE ACTIVITY: ---
<!-- $L0=EIMAD HRF7720 040211 PDMKL1 EIM Auditing @L0A---
<!-- $P1=EIMAD HRF7720 040517 PDMKL1 MG03855 @P1A---
<!-- $P2=EIMAD HRF7720 040517 PDMKL1 MG03918 @P2A---
<!-- $P3=EIMAD HRF7720 040518 PDMKL1 MG03931 @P3A---
<!-- $P4=EIMAD HRF7720 040524 PDMKL1 MG03878 @P4A---
<!-- $P5=EIMAD HRF7720 040524 PDMKL1 MG03979 @P5A---
<!-- $P6=EIMAD HRF7720 040601 PDMKL1 MG03993 @P6A---
<!-- $P7=EIMAD HRF7720 040608 PDMKL1 MG03929 @P7A---
<!-- $P8=EIMAD HRF7720 040608 PDMKL1 MG04043 @P8A---
<!-- $P9=EIMAD HRF7720 040625 PDMKL1 MG04193 @P9A---
<!-- $L1=PHRS HRF7730 050404 PDAWS1 Pass Phrase Support @L1A---
<!-- $L2=PKIS7 HRF7730 050503 PDPFW PKIS7 @L2A---
<!-- $O1=OA11912 HRF7720 050524 PDGTM1 Roll-Up APAR OA09052 @O1A---
<!-- $L3=SAFID HRF7730 050524 PDAWS1 SAF Identity Token @L3A---
<!-- $PA=PKIS7 HRF7730 050914 PDMKL1 Defect MG06085 @PAA---
<!-- $L4=PKIS9 HRF7740 060606 PDRDC1 PKIS9 @L4A---
<!-- $L5=SQLROLE HRF7740 060626 PDAWS1 DB2 Support @L5A---
<!-- $PB=PKCS11 HRF7740 060628 PDJJP1 PKCS11 Support @PBA---
<!-- $PC=MG07744 HRF7740 060811 PDRDC1 Rename HighTrust @PCA---
<!-- $L6=CFIELD2 HRF7750 070605 PDAWS1 Custom Field Support @L6A---
<!-- $L7=PKIS11K HRF7760 080701 PDALF1 PKI Key Generation III @L7A---
<!-- $L8=FIPS HRF7760 080523 PDXS1 SIGVER support @L8A---
<!-- $L9=FIPS HRF7760 080716 PDXS1 SIGVER support @L9A---
<!-- $LA=AUTOUID HRF7760 080717 PDXS1 Auto UID/GID @LAA---
<!-- $PD=MG11385 HRF7760 080826 PDALF1 Defect MG11385 @PDA---
<!-- $PE=MG11382 HRF7760 080902 PDALF1 Defect MG11382 @PEA---
<!-- $PF=MG11753 HRF7760 081208 PDAWS1 Defect MG11753 @PFA---
<!-- $PG=MG12241 HRF7760 090218 PDJCL1 Defect MG12241 @PGA---
<!-- $PH=MG12336 HRF7760 090311 PDJLW1 Defect MG12336 @PHA---
<!-- ---
<!-- ---
<!-- CHANGE DESCRIPTIONS: ---
<!-- A000000-999999 @L0A---
<!-- C - Updated the length for utkExecnode, utkNetw, @P1A---
<!-- utkSesstype, and utkSnode. Updated type for @P1A---
<!-- timeWritten to include hundredths of a second. @P1A---
<!-- Refined definition of tokens. @P1A---
<!-- A - Added pwdMixed. @P2A---
<!-- A - Added eventLink. @P3A---
<!-- C - Customers may change smfUserID so it may not be @P4A---
<!-- a RACF userID. Add logRauditx and prodId. @P4A---
<!-- D - Removed link. @P5A---
<!-- D - Removed name. @P6A---
<!-- M - prodName should appear after prodFMID. @P7A---
<!-- C - Make prodName optional. @P8A---
<!-- A - Added response and nestPrimary. @P9A---
<!-- A - Added newPhrExit @L1A---
<!-- A - Added caDomain @L2A---

```

```

<!-- A - Support for z/OS Multi Level Security -->
<!-- - Roll-Up Common Criteria APAR OA09052 which -->
<!-- added the element resSec1 (Resource SECLABEL) @01A-->
<!-- A - Support for SAF Identity Token -->
<!-- - Add ctxUser, ctxReg, ctxHost and ctxMech -->
<!-- - Add new type definitions for t_token1_16, -->
<!-- t_string1_128 and t_string1_510 @L3A-->
<!-- C - Change the syntax for elements creationDate, @PAA-->
<!-- lastModDate, notafterDate, notbeforDate @PAA-->
<!-- A - Added attrReuse, attrTrust, attrHighTrust, @L4A-->
<!-- attrDelete, certUsage, certDefault, privateKey, @L4A-->
<!-- exitPath @L4A-->
<!-- A - Support for DB2 V9 -->
<!-- - Add criteria and new type definition of -->
<!-- t_string1_244 @L5A-->
<!-- A - Support for PKCS11 @PBA-->
<!-- A - Rename attrHighTrust attrHiTrust @PCA-->
<!-- A - Added fldValExit for Custom Fields @L6A-->
<!-- A - Support for PKI Key Generation @L7A-->
<!-- A - Support for SIGVER @L8A-->
<!-- A - Added expirDate for SIGVER @L9A-->
<!-- A - Support for AUTO UID/GID @LAA-->
<!-- C - Change field names to conform to XML standards @PDA-->
<!-- C - Change keySz to keySize @PEA-->
<!-- C - Changed definition of smfUserId to t_string1_16 @PFA-->
<!-- A - Add ididUser and ididReg for IDPROP @PGA-->
<!-- D - Changed to licensed material. @PHA-->
<!-- -->
<!-- ----- -->

<!-- ----- -->
<!-- Schema definitions for product/component -->
<!-- specific event details. -->
<!-- ----- -->

<xs:import namespace="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  schemaLocation="dcmes-rdf.xsd"/>
<xs:import namespace="http://www.ibm.com/xmlns/zOS/EIMSchema"
  schemaLocation="irreimsc.xsd"/>

<!-- ----- -->
<!-- RACF base types -->
<!-- ----- -->

<xs:simpleType name="t_dataset">
  <xs:restriction base="xs:token">
    <xs:pattern value="\S+"/>
    <xs:minLength value="1"/>
    <xs:maxLength value="44"/>
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="t_date">
  <xs:restriction base="xs:date">
    <xs:pattern value="\d\d\d\d-\d\d?-\d\d?"/>
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="t_dateSlashes">
  <xs:restriction base="xs:string">
    <xs:pattern value="\d\d\d\d/\d\d?/\d\d?"/>
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="t_fmId">
  <xs:restriction base="xs:token">
    <xs:pattern value="\S+"/>

```

```

        <xs:length value="7"/>
    </xs:restriction>
</xs:simpleType>

<xs:simpleType name="t_gid">
    <xs:restriction base="xs:token">
        <xs:pattern value="\S+"/>
        <xs:length value="10"/>
    </xs:restriction>
</xs:simpleType>

<xs:simpleType name="t_time">
    <xs:restriction base="xs:token">
        <xs:pattern value="\d\d?:\d\d?:\d\d?"/>
    </xs:restriction>
</xs:simpleType>

<xs:simpleType name="t_timex">
    <xs:restriction base="xs:token">
        <xs:pattern value="\d\d?:\d\d?:\d\d?.\d\d"/>
    </xs:restriction>
</xs:simpleType>

<xs:simpleType name="t_uid">
    <xs:restriction base="xs:token">
        <xs:pattern value="\S+"/>
        <xs:length value="10"/>
    </xs:restriction>
</xs:simpleType>

<xs:simpleType name="t_unit">
    <xs:restriction base="xs:token">
        <xs:pattern value="\S+"/>
        <xs:minLength value="1"/>
        <xs:maxLength value="6"/>
    </xs:restriction>
</xs:simpleType>

<xs:simpleType name="t_userid">
    <xs:restriction base="xs:token">
        <xs:pattern value="\S+"/>
        <xs:minLength value="1"/>
        <xs:maxLength value="8"/>
    </xs:restriction>
</xs:simpleType>

<xs:simpleType name="t_vol">
    <xs:restriction base="xs:token">
        <xs:pattern value="\S+"/>
        <xs:minLength value="1"/>
        <xs:maxLength value="6"/>
    </xs:restriction>
</xs:simpleType>

<xs:simpleType name="t_yesNo">
    <xs:restriction base="xs:token">
        <xs:enumeration value="Y"/>
        <xs:enumeration value="N"/>
    </xs:restriction>
</xs:simpleType>

<xs:simpleType name="t_integer1">
    <xs:restriction base="xs:nonNegativeInteger">
        <xs:maxInclusive value="9"/>
    </xs:restriction>
</xs:simpleType>

```



```

<xs:simpleType name="t_string1_4">
  <xs:restriction base="xs:string">
    <xs:minLength value="1"/>
    <xs:maxLength value="4"/>
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="t_string1_8">
  <xs:restriction base="xs:string">
    <xs:minLength value="1"/>
    <xs:maxLength value="8"/>
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="t_string1_10">
  <xs:restriction base="xs:string">
    <xs:minLength value="1"/>
    <xs:maxLength value="10"/>
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="t_string1_13">
  <xs:restriction base="xs:string">
    <xs:minLength value="1"/>
    <xs:maxLength value="13"/>
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="t_string1_16">
  <xs:restriction base="xs:string">
    <xs:minLength value="1"/>
    <xs:maxLength value="16"/>
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="t_string1_20">
  <xs:restriction base="xs:string">
    <xs:minLength value="1"/>
    <xs:maxLength value="20"/>
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="t_string1_22">
  <xs:restriction base="xs:string">
    <xs:minLength value="1"/>
    <xs:maxLength value="22"/>
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="t_string1_30">
  <xs:restriction base="xs:string">
    <xs:length value="30"/>
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="t_string1_32">
  <xs:restriction base="xs:string">
    <xs:minLength value="1"/>
    <xs:maxLength value="32"/>
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="t_string1_36">
  <xs:restriction base="xs:string">
    <xs:minLength value="1"/>
    <xs:maxLength value="36"/>
  </xs:restriction>

```

```

</xs:simpleType>

<xs:simpleType name="t_string1_40">
  <xs:restriction base="xs:string">
    <xs:minLength value="1"/>
    <xs:maxLength value="40"/>
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="t_string1_45">
  <xs:restriction base="xs:string">
    <xs:minLength value="1"/>
    <xs:maxLength value="45"/>
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="t_string1_56">
  <xs:restriction base="xs:string">
    <xs:minLength value="1"/>
    <xs:maxLength value="56"/>
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="t_string1_64">
  <xs:restriction base="xs:string">
    <xs:minLength value="1"/>
    <xs:maxLength value="64"/>
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="t_string1_100">
  <xs:restriction base="xs:string">
    <xs:minLength value="1"/>
    <xs:maxLength value="100"/>
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="t_string1_128">
  <xs:restriction base="xs:string">
    <xs:minLength value="1"/>
    <xs:maxLength value="128"/>
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="t_string1_240">
  <xs:restriction base="xs:string">
    <xs:minLength value="1"/>
    <xs:maxLength value="240"/>
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="t_string1_244">
  <xs:restriction base="xs:string">
    <xs:minLength value="1"/>
    <xs:maxLength value="244"/>
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="t_string1_246">
  <xs:restriction base="xs:string">
    <xs:minLength value="1"/>
    <xs:maxLength value="246"/>
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="t_string1_255">
  <xs:restriction base="xs:string">

```

```

        <xs:minLength value="1"/>
        <xs:maxLength value="255"/>
    </xs:restriction>
</xs:simpleType>

<xs:simpleType name="t_string1_256">
    <xs:restriction base="xs:string">
        <xs:minLength value="1"/>
        <xs:maxLength value="256"/>
    </xs:restriction>
</xs:simpleType>

<xs:simpleType name="t_string1_510">
    <xs:restriction base="xs:string">
        <xs:minLength value="1"/>
        <xs:maxLength value="510"/>
    </xs:restriction>
</xs:simpleType>

<xs:simpleType name="t_string1_985">
    <xs:restriction base="xs:string">
        <xs:minLength value="1"/>
        <xs:maxLength value="985"/>
    </xs:restriction>
</xs:simpleType>

<xs:simpleType name="t_string1_1021">
    <xs:restriction base="xs:string">
        <xs:minLength value="1"/>
        <xs:maxLength value="1021"/>
    </xs:restriction>
</xs:simpleType>

<xs:simpleType name="t_string1_1023">
    <xs:restriction base="xs:string">
        <xs:minLength value="1"/>
        <xs:maxLength value="1023"/>
    </xs:restriction>
</xs:simpleType>

<xs:simpleType name="t_string1_1024">
    <xs:restriction base="xs:string">
        <xs:minLength value="1"/>
        <xs:maxLength value="1024"/>
    </xs:restriction>
</xs:simpleType>

<xs:simpleType name="t_string1_4000">
    <xs:restriction base="xs:string">
        <xs:minLength value="1"/>
        <xs:maxLength value="4000"/>
    </xs:restriction>
</xs:simpleType>

<xs:simpleType name="t_string1_4096">
    <xs:restriction base="xs:string">
        <xs:minLength value="1"/>
        <xs:maxLength value="4096"/>
    </xs:restriction>
</xs:simpleType>

<xs:simpleType name="t_token3">
    <xs:restriction base="xs:token">
        <xs:pattern value="\S+"/>
        <xs:length value="3"/>
    </xs:restriction>
</xs:simpleType>

```

```

<xs:simpleType name="t_token1_3">
  <xs:restriction base="xs:token">
    <xs:pattern value="\S+"/>
    <xs:minLength value="1"/>
    <xs:maxLength value="3"/>
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="t_token4">
  <xs:restriction base="xs:token">
    <xs:pattern value="\S+"/>
    <xs:length value="4"/>
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="t_token1_4">
  <xs:restriction base="xs:token">
    <xs:pattern value="\S+"/>
    <xs:minLength value="1"/>
    <xs:maxLength value="4"/>
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="t_token5">
  <xs:restriction base="xs:token">
    <xs:pattern value="\S+"/>
    <xs:length value="5"/>
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="t_token1_6">
  <xs:restriction base="xs:token">
    <xs:pattern value="\S+"/>
    <xs:minLength value="1"/>
    <xs:maxLength value="6"/>
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="t_token1_7">
  <xs:restriction base="xs:token">
    <xs:pattern value="\S+"/>
    <xs:minLength value="1"/>
    <xs:maxLength value="7"/>
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="t_token1_8">
  <xs:restriction base="xs:token">
    <xs:pattern value="\S+"/>
    <xs:minLength value="1"/>
    <xs:maxLength value="8"/>
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="t_token10">
  <xs:restriction base="xs:token">
    <xs:pattern value="\S+"/>
    <xs:length value="10"/>
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="t_token1_11">
  <xs:restriction base="xs:token">
    <xs:pattern value="\S+"/>
    <xs:minLength value="1"/>
    <xs:maxLength value="11"/>
  </xs:restriction>
</xs:simpleType>

```

```

    </xs:restriction>
</xs:simpleType>

<xs:simpleType name="t_token1_16">
  <xs:restriction base="xs:token">
    <xs:pattern value="\S+"/>
    <xs:minLength value="1"/>
    <xs:maxLength value="16"/>
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="t_token32">
  <xs:restriction base="xs:token">
    <xs:pattern value="\S+"/>
    <xs:length value="32"/>
  </xs:restriction>
</xs:simpleType>

<!-- - - - - - -->
<!-- Beginning of the security event log element defintions -->
<!-- - - - - - -->
<xs:element name="securityEventLog">
  <xs:complexType>
    <xs:sequence>

      <!-- meta information about this security event log -->
      <!-- <rdf:Description>...</rdf:Description> -->
      <xs:any namespace="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
        processContents="strict"/>

      <!-- <event>...</event> -->
      <xs:element name="event" minOccurs="1" maxOccurs="unbounded">
        <xs:complexType>
          <xs:sequence>
            <!-- -->
            <!-- Common elements of all records -->
            <!-- -->
            <xs:element name="eventType" type="t_token1_8" />
            <xs:element name="eventQual" type="t_token1_8"
              minOccurs="0" maxOccurs="1"/>
            <xs:element name="timeWritten" type="t_timex" />
            <xs:element name="dateWritten" type="t_date" />
            <xs:element name="systemSmfid" type="t_token1_4" />
            <xs:element name="prodFmid" type="t_fmfd"
              minOccurs="0" maxOccurs="1"/>
            <xs:element name="prodName" type="t_string1_255"
              minOccurs="0" maxOccurs="1"/>

          <xs:element name="details">
            <xs:complexType>
              <xs:choice minOccurs="1" maxOccurs="unbounded">

                <xs:element name="accessType" type="t_token1_8" />
                <xs:element name="aceeCompress" type="t_yesNo" />
                <xs:element name="aceeCompXm" type="t_yesNo" />
                <xs:element name="acIType" type="t_token1_8" />
                <xs:element name="action" type="t_string1_16" />
                <xs:element name="actionCom" type="t_string1_64" />
                <xs:element name="active" type="t_yesNo" />
                <xs:element name="addcreator" type="t_yesNo" />
                <xs:element name="adsp" type="t_yesNo" />
                <xs:element name="allowedExec" type="t_yesNo" />
                <xs:element name="allowedRead" type="t_yesNo" />
                <xs:element name="allowedWrite" type="t_yesNo" />
                <xs:element name="allCmdExit" type="t_yesNo" />
                <xs:element name="altDomain" type="t_string1_100" />
                <xs:element name="altEmail" type="t_string1_100" />
              </xs:choice>
            </xs:complexType>
          </xs:element>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>

```

```

<xs:element name="altIp" type="t_string1_64" />
<xs:element name="altOther" type="t_string1_1024"/>
<xs:element name="altUri" type="t_string1_255" />
<xs:element name="anewExec" type="t_token1_8" />
<xs:element name="anewRead" type="t_token1_8" />
<xs:element name="anewWrite" type="t_token1_8" />
<xs:element name="aoldExec" type="t_token1_8" />
<xs:element name="aoldRead" type="t_token1_8" />
<xs:element name="aoldWrite" type="t_token1_8" />
<xs:element name="appcLink" type="t_string1_16" />
<xs:element name="appl" type="t_token1_8" />
<xs:element name="applaud" type="t_yesNo" />
<xs:element name="application" type="t_token1_8" />
<xs:element name="assocStatus" type="t_token1_8" />
<xs:element name="attrDelete" type="t_yesNo" />
<xs:element name="attrHiTrust" type="t_yesNo" />
<xs:element name="attrReuse" type="t_yesNo" />
<xs:element name="attrTrust" type="t_yesNo" />
<xs:element name="audit" type="t_yesNo" />
<xs:element name="auditCmdviol" type="t_yesNo" />
<xs:element name="auditCode" type="t_token1_11" />
<xs:element name="auditDasdvol" type="t_yesNo" />
<xs:element name="auditDataset" type="t_yesNo" />
<xs:element name="auditGroup" type="t_yesNo" />
<xs:element name="auditLevel" type="t_yesNo" />
<xs:element name="auditOper" type="t_yesNo" />
<xs:element name="auditSpecial" type="t_yesNo" />
<xs:element name="auditTapevol" type="t_yesNo" />
<xs:element name="auditTerm" type="t_yesNo" />
<xs:element name="auditUser" type="t_yesNo" />
<xs:element name="authinfoacc" type="t_string1_1024"/>
<xs:element name="authAudit" type="t_yesNo" />
<xs:element name="authBypass" type="t_yesNo" />
<xs:element name="authExit" type="t_yesNo" />
<xs:element name="authFailsft" type="t_yesNo" />
<xs:element name="authNormal" type="t_yesNo" />
<xs:element name="authOmvssu" type="t_yesNo" />
<xs:element name="authOmvssys" type="t_yesNo" />
<xs:element name="authOper" type="t_yesNo" />
<xs:element name="authSpecial" type="t_yesNo" />
<xs:element name="authTrusted" type="t_yesNo" />
<xs:element name="authType" type="t_string1_13" />
<xs:element name="backoutFail" type="t_yesNo" />
<xs:element name="badJobname" type="t_token1_8" />
<xs:element name="batchallracf" type="t_yesNo" />
<xs:element name="caDomain" type="t_string1_8" />
<xs:element name="catdsns" type="t_yesNo" />
<xs:element name="catdsnsFail" type="t_yesNo" />
<xs:element name="caudRequestExec" type="t_token1_8" />
<xs:element name="caudRequestRead" type="t_token1_8" />
<xs:element name="caudRequestWrite" type="t_token1_8" />
<xs:element name="certpolicies" type="t_string1_32" />
<xs:element name="certDefault" type="t_yesNo" />
<xs:element name="certDs" type="t_dataset" />
<xs:element name="certId" type="t_string1_56" />
<xs:element name="certUsage" type="t_token1_8" />
<xs:element name="class" type="t_token1_8" />
<xs:element name="className" type="t_token1_8" />
<xs:element name="cmdExit" type="t_yesNo" />
<xs:element name="compatmode" type="t_yesNo" />
<xs:element name="creationDate" type="t_dateSlashes" />
<xs:element name="creatorGid" type="t_gid" />
<xs:element name="creatorUid" type="t_uid" />
<xs:element name="credType" type="t_string1_30" />
<xs:element name="criteria" type="t_string1_244" />
<xs:element name="critical" type="t_string1_255" />
<xs:element name="crlSerNum" type="t_string1_255" />

```

```

<xs:element name="ctxHost" type="t_string1_128" />
<xs:element name="ctxMech" type="t_token1_16" />
<xs:element name="ctxReg" type="t_string1_255" />
<xs:element name="ctxUser" type="t_string1_510" />
<xs:element name="dasd" type="t_yesNo" />
<xs:element name="dasdStats" type="t_yesNo" />
<xs:element name="datasetName" type="t_dataset" />
<xs:element name="datasetStats" type="t_yesNo" />
<xs:element name="datasetUnit" type="t_token1_3" />
<xs:element name="datasetVol" type="t_vol" />
<xs:element name="dataSet" type="t_dataset" />
<xs:element name="dceLink" type="t_string1_16" />
<xs:element name="delCmdExit" type="t_yesNo" />
<xs:element name="dfltProcess" type="t_yesNo" />
<xs:element name="dftPri" type="t_token1_3" />
<xs:element name="dftSec" type="t_token1_3" />
<xs:element name="dsName" type="t_dataset" />
<xs:element name="dupDsns" type="t_yesNo" />
<xs:element name="earlyverify" type="t_yesNo" />
<xs:element name="egn" type="t_yesNo" />
<xs:element name="encryptExit" type="t_yesNo" />
<xs:element name="encryptExit2" type="t_yesNo" />
<xs:element name="entryId" type="t_string1_10" />
<xs:element name="entryType" type="t_yesNo" />
<xs:element name="erase" type="t_yesNo" />
<xs:element name="eraseAll" type="t_yesNo" />
<xs:element name="eraseLevel" type="t_yesNo" />
<xs:element name="eventLink" type="t_string1_16" />
<xs:element name="evtGrpId" type="t_token1_8" />
<xs:element name="evtUserId" type="t_userid" />
<xs:element name="exitPath" type="t_string1_256" />
<xs:element name="expirDate" type="t_date" />
<xs:element name="extkeyusage" type="t_string1_255" />
<xs:element name="failed" type="t_string1_1024"/>
<xs:element name="fastauthPost" type="t_yesNo" />
<xs:element name="fastauthPre" type="t_yesNo" />
<xs:element name="filepool" type="t_token1_8" />
<xs:element name="filepool2" type="t_token1_8" />
<xs:element name="filespace" type="t_token1_8" />
<xs:element name="filespace2" type="t_token1_8" />
<xs:element name="fileId" type="t_string1_32" />
<xs:element name="fileId2" type="t_string1_32" />
<xs:element name="fileName" type="t_string1_256" />
<xs:element name="fileOwnGid" type="t_gid" />
<xs:element name="fileOwnUid" type="t_uid" />
<xs:element name="file1Id" type="t_token32" />
<xs:element name="file1OwnGid" type="t_gid" />
<xs:element name="file1OwnUid" type="t_uid" />
<xs:element name="file2Id" type="t_token32" />
<xs:element name="file2OwnGid" type="t_gid" />
<xs:element name="file2OwnUid" type="t_uid" />
<xs:element name="fldValExit" type="t_yesNo" />
<xs:element name="fracheckPost" type="t_yesNo" />
<xs:element name="fracheckPre" type="t_yesNo" />
<xs:element name="gencmd" type="t_yesNo" />
<xs:element name="generic" type="t_yesNo" />
<xs:element name="genericOwner" type="t_yesNo" />
<xs:element name="genlist" type="t_yesNo" />
<xs:element name="gid" type="t_gid" />
<xs:element name="global" type="t_yesNo" />
<xs:element name="grant" type="t_token1_8" />
<xs:element name="grplist" type="t_yesNo" />
<xs:element name="grpId" type="t_token1_8" />
<xs:element name="hfsDsName" type="t_dataset" />
<xs:element name="hostidMap" type="t_string1_1024"/>
<xs:element name="hostUserId" type="t_userid" />
<xs:element name="id" type="t_token10" />

```

```

<xs:element name="ididReg" type="t_string1_1021"/>
<xs:element name="ididUser" type="t_string1_985" />
<xs:element name="ignored" type="t_string1_1024"/>
<xs:element name="inactive" type="t_token3" />
<xs:element name="inode" type="t_token10" />
<xs:element name="inode2" type="t_token10" />
<xs:element name="issuersDn" type="t_string1_255" />
<xs:element name="issueId" type="t_token1_8" />
<xs:element name="issueNode" type="t_token1_8" />
<xs:element name="issuingDpDn" type="t_string1_255" />
<xs:element name="issuingUri" type="t_string1_1024"/>
<xs:element name="jobName" type="t_token1_8" />
<xs:element name="kdcStatCode" type="t_string1_10" />
<xs:element name="kerblvl" type="t_token3" />
<xs:element name="kerbPrincipal" type="t_string1_240" />
<xs:element name="key" type="t_token1_8" />
<xs:element name="keyusage" type="t_string1_64" />
<xs:element name="keyId" type="t_string1_40" />
<xs:element name="keyOwnGid" type="t_gid" />
<xs:element name="keyOwnUid" type="t_uid" />
<xs:element name="keySize" type="t_string1_4" />
<xs:element name="lastDeleted" type="t_yesNo" />
<xs:element name="lastModDate" type="t_dateSlashes" />
<xs:element name="level" type="t_token3" />
<xs:element name="levelAudit" type="t_token5" />
<xs:element name="levelErase" type="t_token5" />
<xs:element name="loginSource" type="t_string1_22" />
<xs:element name="logstr" type="t_string1_255" />
<xs:element name="logAccess" type="t_yesNo" />
<xs:element name="logAlways" type="t_yesNo" />
<xs:element name="logApplaud" type="t_yesNo" />
<xs:element name="logClass" type="t_yesNo" />
<xs:element name="logCmdviol" type="t_yesNo" />
<xs:element name="logCompatm" type="t_yesNo" />
<xs:element name="logGlobal" type="t_yesNo" />
<xs:element name="logLevel" type="t_yesNo" />
<xs:element name="logLogopt" type="t_yesNo" />
<xs:element name="logNonomvs" type="t_yesNo" />
<xs:element name="logOmvsnprv" type="t_yesNo" />
<xs:element name="logOptions" type="t_string1_8" />
<xs:element name="logRacinit" type="t_yesNo" />
<xs:element name="logRauditx" type="t_yesNo" />
<xs:element name="logSec1" type="t_yesNo" />
<xs:element name="logSpecial" type="t_yesNo" />
<xs:element name="logUser" type="t_yesNo" />
<xs:element name="logVmevent" type="t_yesNo" />
<xs:element name="mlactive" type="t_yesNo" />
<xs:element name="mlactiveFail" type="t_yesNo" />
<xs:element name="mlfs" type="t_token1_8" />
<xs:element name="mlipc" type="t_token1_8" />
<xs:element name="mlnames" type="t_yesNo" />
<xs:element name="mlquiet" type="t_yesNo" />
<xs:element name="mls" type="t_yesNo" />
<xs:element name="mlstable" type="t_yesNo" />
<xs:element name="mlsFail" type="t_yesNo" />
<xs:element name="modClass" type="t_string1_8" />
<xs:element name="modData" type="t_string1_4000"/>
<xs:element name="modelGdg" type="t_yesNo" />
<xs:element name="modelGroup" type="t_yesNo" />
<xs:element name="modelName" type="t_string1_255" />
<xs:element name="modelUser" type="t_yesNo" />
<xs:element name="modelVol" type="t_vol" />
<xs:element name="modLoaded" type="t_yesNo" />
<xs:element name="modProf" type="t_string1_255" />
<xs:element name="modService" type="t_string1_20" />
<xs:element name="namingConv" type="t_yesNo" />
<xs:element name="nestPrimary" type="t_userid" />

```



```

<xs:element name="newsecl" type="t_token1_8" />
<xs:element name="newEffGid" type="t_gid" />
<xs:element name="newEffUid" type="t_uid" />
<xs:element name="newExecute" type="t_yesNo" />
<xs:element name="newGrpExec" type="t_yesNo" />
<xs:element name="newGrpRead" type="t_yesNo" />
<xs:element name="newGrpWrite" type="t_yesNo" />
<xs:element name="newOthExec" type="t_yesNo" />
<xs:element name="newOthRead" type="t_yesNo" />
<xs:element name="newOthWrite" type="t_yesNo" />
<xs:element name="newOwnExec" type="t_yesNo" />
<xs:element name="newOwnRead" type="t_yesNo" />
<xs:element name="newOwnWrite" type="t_yesNo" />
<xs:element name="newPhrExit" type="t_yesNo" />
<xs:element name="newPwdExit" type="t_yesNo" />
<xs:element name="newRead" type="t_yesNo" />
<xs:element name="newRealGid" type="t_gid" />
<xs:element name="newRealUid" type="t_uid" />
<xs:element name="newResName" type="t_string1_255" />
<xs:element name="newSavedGid" type="t_gid" />
<xs:element name="newSavedUid" type="t_uid" />
<xs:element name="newSIsgid" type="t_yesNo" />
<xs:element name="newSIsuid" type="t_yesNo" />
<xs:element name="newSIsvtx" type="t_yesNo" />
<xs:element name="newWrite" type="t_yesNo" />
<xs:element name="nextDate" type="t_date" />
<xs:element name="nextTime" type="t_time" />
<xs:element name="njeNameId" type="t_token1_8" />
<xs:element name="njeUdfndId" type="t_token1_8" />
<xs:element name="noauthClauth" type="t_yesNo" />
<xs:element name="noauthGroup" type="t_yesNo" />
<xs:element name="noauthProf" type="t_yesNo" />
<xs:element name="notafterDate" type="t_dateSlashes" />
<xs:element name="notbeforDate" type="t_dateSlashes" />
<xs:element name="notifyEmail" type="t_string1_64" />
<xs:element name="object" type="t_string1_4096" />
<xs:element name="oldvol" type="t_vol" />
<xs:element name="oldEffGid" type="t_gid" />
<xs:element name="oldEffUid" type="t_uid" />
<xs:element name="oldExecute" type="t_yesNo" />
<xs:element name="oldGrpExec" type="t_yesNo" />
<xs:element name="oldGrpRead" type="t_yesNo" />
<xs:element name="oldGrpWrite" type="t_yesNo" />
<xs:element name="oldOthExec" type="t_yesNo" />
<xs:element name="oldOthRead" type="t_yesNo" />
<xs:element name="oldOthWrite" type="t_yesNo" />
<xs:element name="oldOwnExec" type="t_yesNo" />
<xs:element name="oldOwnRead" type="t_yesNo" />
<xs:element name="oldOwnWrite" type="t_yesNo" />
<xs:element name="oldRead" type="t_yesNo" />
<xs:element name="oldRealGid" type="t_gid" />
<xs:element name="oldRealUid" type="t_uid" />
<xs:element name="oldSavedGid" type="t_gid" />
<xs:element name="oldSavedUid" type="t_uid" />
<xs:element name="oldSec1" type="t_token1_8" />
<xs:element name="oldSIsgid" type="t_yesNo" />
<xs:element name="oldSIsuid" type="t_yesNo" />
<xs:element name="oldSIsvtx" type="t_yesNo" />
<xs:element name="oldWrite" type="t_yesNo" />
<xs:element name="optype" type="t_token1_8" />
<xs:element name="ownerGid" type="t_gid" />
<xs:element name="ownerUid" type="t_uid" />
<xs:element name="ownId" type="t_token1_8" />
<xs:element name="passPhrase" type="t_yesNo" />
<xs:element name="pathName" type="t_string1_1023" />
<xs:element name="pathType" type="t_token1_4" />
<xs:element name="path2" type="t_string1_1023" />

```

```

<xs:element name="pdasPrincipal" type="t_string1_36" />
<xs:element name="pdsDsn" type="t_dataset" />
<xs:element name="phase" type="t_string1_20" />
<xs:element name="pkdsLabel" type="t_string1_64" />
<xs:element name="prevSerial" type="t_string1_255" />
<xs:element name="privateKey" type="t_yesNo" />
<xs:element name="prodId" type="t_string1_8" />
<xs:element name="profileName" type="t_string1_246" />
<xs:element name="profSame" type="t_yesNo" />
<xs:element name="protectall" type="t_yesNo" />
<xs:element name="protectallw" type="t_yesNo" />
<xs:element name="publishDate" type="t_date" />
<xs:element name="publishTime" type="t_time" />
<xs:element name="pwdMixed" type="t_yesNo" />
<xs:element name="pwdrule1" type="t_string1_8" />
<xs:element name="pwdrule1Max" type="t_integer1" />
<xs:element name="pwdrule1Min" type="t_integer1" />
<xs:element name="pwdrule2" type="t_string1_8" />
<xs:element name="pwdrule2Max" type="t_integer1" />
<xs:element name="pwdrule2Min" type="t_integer1" />
<xs:element name="pwdrule3" type="t_string1_8" />
<xs:element name="pwdrule3Max" type="t_integer1" />
<xs:element name="pwdrule3Min" type="t_integer1" />
<xs:element name="pwdrule4" type="t_string1_8" />
<xs:element name="pwdrule4Max" type="t_integer1" />
<xs:element name="pwdrule4Min" type="t_integer1" />
<xs:element name="pwdrule5" type="t_string1_8" />
<xs:element name="pwdrule5Max" type="t_integer1" />
<xs:element name="pwdrule5Min" type="t_integer1" />
<xs:element name="pwdrule6" type="t_string1_8" />
<xs:element name="pwdrule6Max" type="t_integer1" />
<xs:element name="pwdrule6Min" type="t_integer1" />
<xs:element name="pwdrule7" type="t_string1_8" />
<xs:element name="pwdrule7Max" type="t_integer1" />
<xs:element name="pwdrule7Min" type="t_integer1" />
<xs:element name="pwdrule8" type="t_string1_8" />
<xs:element name="pwdrule8Max" type="t_integer1" />
<xs:element name="pwdrule8Min" type="t_integer1" />
<xs:element name="pwdHist" type="t_token3" />
<xs:element name="pwdInt" type="t_token3" />
<xs:element name="pwdMin" type="t_token3" />
<xs:element name="pwdRevoke" type="t_token3" />
<xs:element name="pwdStatus" type="t_token1_8" />
<xs:element name="pwdWarn" type="t_token3" />
<xs:element name="qop" type="t_token10" />
<xs:element name="racdefPost" type="t_yesNo" />
<xs:element name="racdefPre" type="t_yesNo" />
<xs:element name="racfVersion" type="t_token4" />
<xs:element name="racheckPost" type="t_yesNo" />
<xs:element name="racheckPre" type="t_yesNo" />
<xs:element name="racinitPost" type="t_yesNo" />
<xs:element name="racinitPre" type="t_yesNo" />
<xs:element name="racinitStats" type="t_yesNo" />
<xs:element name="raclist" type="t_yesNo" />
<xs:element name="raclistPre" type="t_yesNo" />
<xs:element name="raclistSel" type="t_yesNo" />
<xs:element name="readDate" type="t_date" />
<xs:element name="readTime" type="t_time" />
<xs:element name="realsdn" type="t_yesNo" />
<xs:element name="recvr" type="t_token1_8" />
<xs:element name="request" type="t_token1_8" />
<xs:element name="requestor" type="t_string1_32" />
<xs:element name="requestorEmail" type="t_string1_32" />
<xs:element name="requestDsrch" type="t_yesNo" />
<xs:element name="requestExec" type="t_yesNo" />
<xs:element name="requestPath2" type="t_string1_1023" />
<xs:element name="requestRead" type="t_yesNo" />

```

```

<xs:element name="requestWrite" type="t_yesNo" />
<xs:element name="reqGrpExec" type="t_yesNo" />
<xs:element name="reqGrpRead" type="t_yesNo" />
<xs:element name="reqGrpWrite" type="t_yesNo" />
<xs:element name="reqOthExec" type="t_yesNo" />
<xs:element name="reqOthRead" type="t_yesNo" />
<xs:element name="reqOthWrite" type="t_yesNo" />
<xs:element name="reqOwnExec" type="t_yesNo" />
<xs:element name="reqOwnRead" type="t_yesNo" />
<xs:element name="reqOwnWrite" type="t_yesNo" />
<xs:element name="reqPerms" type="t_string1_1024" />
<xs:element name="reqSIsgid" type="t_yesNo" />
<xs:element name="reqSIsuid" type="t_yesNo" />
<xs:element name="reqSIsvtx" type="t_yesNo" />
<xs:element name="resName" type="t_string1_255" />
<xs:element name="resSec1" type="t_token1_8" />
<xs:element name="response" type="t_string1_1024" />
<xs:element name="retention" type="t_token5" />
<xs:element name="revokeRsn" type="t_string1_32" />
<xs:element name="ringOwner" type="t_userid" />
<xs:element name="riniTerm" type="t_yesNo" />
<xs:element name="rootDn" type="t_string1_255" />
<xs:element name="rstaInstPwd" type="t_yesNo" />
<xs:element name="rswiInstPwd" type="t_yesNo" />
<xs:element name="rtkDefault" type="t_yesNo" />
<xs:element name="rtkDftGrp" type="t_yesNo" />
<xs:element name="rtkDftSec1" type="t_yesNo" />
<xs:element name="rtkEncr" type="t_yesNo" />
<xs:element name="rtkError" type="t_yesNo" />
<xs:element name="rtkExecnode" type="t_token1_8" />
<xs:element name="rtkGrpId" type="t_token1_8" />
<xs:element name="rtkLogusr" type="t_yesNo" />
<xs:element name="rtkNetw" type="t_token1_8" />
<xs:element name="rtkNjeunusr" type="t_yesNo" />
<xs:element name="rtkPre19" type="t_yesNo" />
<xs:element name="rtkPriv" type="t_yesNo" />
<xs:element name="rtkRemote" type="t_yesNo" />
<xs:element name="rtkSec1" type="t_token1_8" />
<xs:element name="rtkSesstype" type="t_token1_8" />
<xs:element name="rtkSgrpId" type="t_token1_8" />
<xs:element name="rtkSnode" type="t_token1_8" />
<xs:element name="rtkSpclass" type="t_token1_8" />
<xs:element name="rtkSpecial" type="t_yesNo" />
<xs:element name="rtkSpoe" type="t_token1_8" />
<xs:element name="rtkSurrogat" type="t_yesNo" />
<xs:element name="rtkSuserId" type="t_userid" />
<xs:element name="rtkTrusted" type="t_yesNo" />
<xs:element name="rtkUnknusr" type="t_yesNo" />
<xs:element name="rtkUserId" type="t_userid" />
<xs:element name="rtkVerprof" type="t_yesNo" />
<xs:element name="scid" type="t_token10" />
<xs:element name="scid2" type="t_token10" />
<xs:element name="secl" type="t_token1_8" />
<xs:element name="seclAudit" type="t_yesNo" />
<xs:element name="seclCtrl" type="t_yesNo" />
<xs:element name="serialNumber" type="t_string1_255" />
<xs:element name="serviceCode" type="t_token1_11" />
<xs:element name="servsecl" type="t_token1_8" />
<xs:element name="servPoename" type="t_string1_64" />
<xs:element name="sessionInt" type="t_token5" />
<xs:element name="signalCode" type="t_token10" />
<xs:element name="signDate" type="t_date" />
<xs:element name="signerDn" type="t_string1_255" />
<xs:element name="signTime" type="t_time" />
<xs:element name="signwith" type="t_string1_45" />
<xs:element name="singleDsn" type="t_token1_8" />
<xs:element name="slbysys" type="t_yesNo" />

```

```

<xs:element name="smfUserId" type="t_string1_16" />
<xs:element name="sourceDate" type="t_date" />
<xs:element name="sourceId" type="t_token1_8" />
<xs:element name="sourceSmfid" type="t_token1_4" />
<xs:element name="sourceTime" type="t_token1_8" />
<xs:element name="specified" type="t_string1_1024"/>
<xs:element name="srvrGrpId" type="t_token1_8" />
<xs:element name="srvrUserId" type="t_userid" />
<xs:element name="stats" type="t_yesNo" />
<xs:element name="status" type="t_string1_32" />
<xs:element name="subjectsDn" type="t_string1_255" />
<xs:element name="symlink" type="t_string1_1023"/>
<xs:element name="symlinkData" type="t_string1_1023"/>
<xs:element name="tapedsn" type="t_yesNo" />
<xs:element name="tapevol" type="t_yesNo" />
<xs:element name="tapevolStats" type="t_yesNo" />
<xs:element name="targetLabel" type="t_string1_32" />
<xs:element name="targetUser" type="t_userid" />
<xs:element name="targetUserId" type="t_userid" />
<xs:element name="term" type="t_token1_8" />
<xs:element name="termLevel" type="t_token3" />
<xs:element name="termNone" type="t_yesNo" />
<xs:element name="termStats" type="t_yesNo" />
<xs:element name="tgtAuthId" type="t_token1_8" />
<xs:element name="tgtEffGid" type="t_gid" />
<xs:element name="tgtEffUid" type="t_uid" />
<xs:element name="tgtId" type="t_token1_8" />
<xs:element name="tgtNode" type="t_token1_8" />
<xs:element name="tgtPid" type="t_token10" />
<xs:element name="tgtRealGid" type="t_gid" />
<xs:element name="tgtRealUid" type="t_uid" />
<xs:element name="tgtSavedGid" type="t_gid" />
<xs:element name="tgtSavedUid" type="t_uid" />
<xs:element name="tgtSavUid" type="t_uid" />
<xs:element name="thisDate" type="t_date" />
<xs:element name="thisTime" type="t_time" />
<xs:element name="token" type="t_string1_32" />
<xs:element name="type" type="t_token1_8" />
<xs:element name="uadsName" type="t_dataset" />
<xs:element name="uadsVol" type="t_vol" />
<xs:element name="uid" type="t_uid" />
<xs:element name="unewExec" type="t_token1_8" />
<xs:element name="unewRead" type="t_token1_8" />
<xs:element name="unewWrite" type="t_token1_8" />
<xs:element name="uoldExec" type="t_token1_8" />
<xs:element name="uoldRead" type="t_token1_8" />
<xs:element name="uoldWrite" type="t_token1_8" />
<xs:element name="usecl" type="t_token1_8" />
<xs:element name="userId" type="t_userid" />
<xs:element name="userName" type="t_string1_20" />
<xs:element name="userNdfnd" type="t_yesNo" />
<xs:element name="userWarning" type="t_yesNo" />
<xs:element name="usrSec1" type="t_token1_8" />
<xs:element name="utkDefault" type="t_yesNo" />
<xs:element name="utkDftGrp" type="t_yesNo" />
<xs:element name="utkDftSec1" type="t_yesNo" />
<xs:element name="utkEncr" type="t_yesNo" />
<xs:element name="utkError" type="t_yesNo" />
<xs:element name="utkExecnode" type="t_token1_8" />
<xs:element name="utkGrpId" type="t_token1_8" />
<xs:element name="utkLogusr" type="t_yesNo" />
<xs:element name="utkNetw" type="t_token1_8" />
<xs:element name="utkNjeunusr" type="t_yesNo" />
<xs:element name="utkPre19" type="t_yesNo" />
<xs:element name="utkPriv" type="t_yesNo" />
<xs:element name="utkRemote" type="t_yesNo" />
<xs:element name="utkRemove" type="t_yesNo" />

```

```

<xs:element name="utkSec1" type="t_token1_8" />
<xs:element name="utkSesstype" type="t_token1_8" />
<xs:element name="utkSgrpId" type="t_token1_8" />
<xs:element name="utkSnode" type="t_token1_8" />
<xs:element name="utkSpclass" type="t_token1_8" />
<xs:element name="utkSpecial" type="t_yesNo" />
<xs:element name="utkSpoe" type="t_token1_8" />
<xs:element name="utkSurrogat" type="t_yesNo" />
<xs:element name="utkSuserId" type="t_userid" />
<xs:element name="utkTrusted" type="t_yesNo" />
<xs:element name="utkUnknusr" type="t_yesNo" />
<xs:element name="utkUserId" type="t_userid" />
<xs:element name="utkVerprof" type="t_yesNo" />
<xs:element name="violation" type="t_yesNo" />
<xs:element name="vol" type="t_vol" />
<xs:element name="whenProgram" type="t_yesNo" />
<xs:element name="xbmallracf" type="t_yesNo" />
<xs:element name="x500Issuer" type="t_string1_255" />
<xs:element name="x500Subject" type="t_string1_255" />

<xs:any namespace="http://www.ibm.com/xmlns/zOS/EIMSchema"
  processContents="strict" />

</xs:choice>
</xs:complexType>
</xs:element> <!-- details element -->
</xs:sequence>
</xs:complexType>
</xs:element> <!-- event element -->
</xs:sequence>
</xs:complexType>
</xs:element> <!-- securityEventLog element -->
</xs:schema>

```

Appendix C. Accessibility

Accessible publications for this product are offered through the z/OS Information Center, which is available at www.ibm.com/systems/z/os/zos/bkserv/.

If you experience difficulty with the accessibility of any z/OS information, please send a detailed message to mhvrcfs@us.ibm.com or to the following mailing address:

IBM Corporation
Attention: MHVRCFS Reader Comments
Department H6MA, Building 707
2455 South Road
Poughkeepsie, NY 12601-5400
USA

Accessibility features

Accessibility features help a user who has a physical disability, such as restricted mobility or limited vision, to use software products successfully. The major accessibility features in z/OS enable users to:

- Use assistive technologies such as screen readers and screen magnifier software
- Operate specific or equivalent features using only the keyboard
- Customize display attributes such as color, contrast, and font size.

Using assistive technologies

Assistive technology products, such as screen readers, function with the user interfaces found in z/OS. Consult the assistive technology documentation for specific information when using such products to access z/OS interfaces.

Keyboard navigation of the user interface

Users can access z/OS user interfaces using TSO/E or ISPF. Refer to *z/OS TSO/E Primer*, *z/OS TSO/E User's Guide*, and *z/OS ISPF User's Guide Vol I* for information about accessing TSO/E and ISPF interfaces. These guides describe how to use TSO/E and ISPF, including the use of keyboard shortcuts or function keys (PF keys). Each guide includes the default settings for the PF keys and explains how to modify their functions.

Dotted decimal syntax diagrams

Syntax diagrams are provided in dotted decimal format for users accessing the z/OS Information Center using a screen reader. In dotted decimal format, each syntax element is written on a separate line. If two or more syntax elements are always present together (or always absent together), they can appear on the same line, because they can be considered as a single compound syntax element.

Each line starts with a dotted decimal number; for example, 3 or 3.1 or 3.1.1. To hear these numbers correctly, make sure that your screen reader is set to read out punctuation. All the syntax elements that have the same dotted decimal number (for example, all the syntax elements that have the number 3.1) are mutually

exclusive alternatives. If you hear the lines 3.1 USERID and 3.1 SYSTEMID, you know that your syntax can include either USERID or SYSTEMID, but not both.

The dotted decimal numbering level denotes the level of nesting. For example, if a syntax element with dotted decimal number 3 is followed by a series of syntax elements with dotted decimal number 3.1, all the syntax elements numbered 3.1 are subordinate to the syntax element numbered 3.

Certain words and symbols are used next to the dotted decimal numbers to add information about the syntax elements. Occasionally, these words and symbols might occur at the beginning of the element itself. For ease of identification, if the word or symbol is a part of the syntax element, it is preceded by the backslash (\) character. The * symbol can be used next to a dotted decimal number to indicate that the syntax element repeats. For example, syntax element *FILE with dotted decimal number 3 is given the format 3 * FILE. Format 3* FILE indicates that syntax element FILE repeats. Format 3* * FILE indicates that syntax element * FILE repeats.

Characters such as commas, which are used to separate a string of syntax elements, are shown in the syntax just before the items they separate. These characters can appear on the same line as each item, or on a separate line with the same dotted decimal number as the relevant items. The line can also show another symbol giving information about the syntax elements. For example, the lines 5.1*, 5.1 LASTRUN, and 5.1 DELETE mean that if you use more than one of the LASTRUN and DELETE syntax elements, the elements must be separated by a comma. If no separator is given, assume that you use a blank to separate each syntax element.

If a syntax element is preceded by the % symbol, this indicates a reference that is defined elsewhere. The string following the % symbol is the name of a syntax fragment rather than a literal. For example, the line 2.1 %OP1 means that you should refer to separate syntax fragment OP1.

The following words and symbols are used next to the dotted decimal numbers:

- ? means an optional syntax element. A dotted decimal number followed by the ? symbol indicates that all the syntax elements with a corresponding dotted decimal number, and any subordinate syntax elements, are optional. If there is only one syntax element with a dotted decimal number, the ? symbol is displayed on the same line as the syntax element, (for example 5? NOTIFY). If there is more than one syntax element with a dotted decimal number, the ? symbol is displayed on a line by itself, followed by the syntax elements that are optional. For example, if you hear the lines 5 ?, 5 NOTIFY, and 5 UPDATE, you know that syntax elements NOTIFY and UPDATE are optional; that is, you can choose one or none of them. The ? symbol is equivalent to a bypass line in a railroad diagram.
- ! means a default syntax element. A dotted decimal number followed by the ! symbol and a syntax element indicates that the syntax element is the default option for all syntax elements that share the same dotted decimal number. Only one of the syntax elements that share the same dotted decimal number can specify a ! symbol. For example, if you hear the lines 2? FILE, 2.1! (KEEP), and 2.1 (DELETE), you know that (KEEP) is the default option for the FILE keyword. In this example, if you include the FILE keyword but do not specify an option, default option KEEP will be applied. A default option also applies to the next higher dotted decimal number. In this example, if the FILE keyword is omitted, default FILE(KEEP) is used. However, if you hear the lines 2? FILE, 2.1, 2.1.1!

(KEEP), and 2.1.1 (DELETE), the default option KEEP only applies to the next higher dotted decimal number, 2.1 (which does not have an associated keyword), and does not apply to 2? FILE. Nothing is used if the keyword FILE is omitted.

- * means a syntax element that can be repeated 0 or more times. A dotted decimal number followed by the * symbol indicates that this syntax element can be used zero or more times; that is, it is optional and can be repeated. For example, if you hear the line 5.1* data area, you know that you can include one data area, more than one data area, or no data area. If you hear the lines 3*, 3 HOST, and 3 STATE, you know that you can include HOST, STATE, both together, or nothing.

Note:

1. If a dotted decimal number has an asterisk (*) next to it and there is only one item with that dotted decimal number, you can repeat that same item more than once.
 2. If a dotted decimal number has an asterisk next to it and several items have that dotted decimal number, you can use more than one item from the list, but you cannot use the items more than once each. In the previous example, you could write HOST STATE, but you could not write HOST HOST.
 3. The * symbol is equivalent to a loop-back line in a railroad syntax diagram.
- + means a syntax element that must be included one or more times. A dotted decimal number followed by the + symbol indicates that this syntax element must be included one or more times; that is, it must be included at least once and can be repeated. For example, if you hear the line 6.1+ data area, you must include at least one data area. If you hear the lines 2+, 2 HOST, and 2 STATE, you know that you must include HOST, STATE, or both. Similar to the * symbol, the + symbol can only repeat a particular item if it is the only item with that dotted decimal number. The + symbol, like the * symbol, is equivalent to a loop-back line in a railroad syntax diagram.

Notices

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

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

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan, Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

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.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

Site Counsel
IBM Corporation
2455 South Road
Poughkeepsie, NY 12601-5400
USA

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

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.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

COPYRIGHT LICENSE:

This information might contain 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. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Policy for unsupported hardware

Various z/OS elements, such as DFSMS, HCD, JES2, JES3, and MVS, contain code that supports specific hardware servers or devices. In some cases, this device-related element support remains in the product even after the hardware devices pass their announced End of Service date. z/OS may continue to service element code; however, it will not provide service related to unsupported hardware devices. Software problems related to these devices will not be accepted

for service, and current service activity will cease if a problem is determined to be associated with out-of-support devices. In such cases, fixes will not be issued.

Minimum supported hardware

The minimum supported hardware for z/OS releases identified in z/OS announcements can subsequently change when service for particular servers or devices is withdrawn. Likewise, the levels of other software products supported on a particular release of z/OS are subject to the service support lifecycle of those products. Therefore, z/OS and its product publications (for example, panels, samples, messages, and product documentation) can include references to hardware and software that is no longer supported.

- For information about software support lifecycle, see: IBM Lifecycle Support for z/OS (<http://www.ibm.com/software/support/systemsz/lifecycle/>)
- For information about currently-supported IBM hardware, contact your IBM representative.

RSA Secure code

This product contains code licensed from RSA Data Security Incorporated.



Trademarks

IBM, the IBM logo, and [ibm.com](http://www.ibm.com) are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml (<http://www.ibm.com/legal/copytrade.shtml>).

Index

Special characters

(OWNER) column heading
in the group tree report 102

A

access attempts
 auditing access attempts to resources
 with security labels 22
 auditing by class 21
 specifying logging for detected 18
access control 1
ACCESS LEVEL column heading
 in the RACF global access checking
 table report 108
accessibility 191
 contact IBM 191
 features 191
accountability 1
activities of group-OPERATIONS
 users 17
activities of OPERATIONS users 17
administration control 16
administration, RACF
 classroom courses x
ALLOWED operand 139
ALTDSD command
 auditing data set accesses 30
 GLOBALAUDIT operand 30
 specifying data set controls 30
ALTER access authority
 in the selected data set report 117
ALTUSER command
 logging RACF-related activities of a
 user 146
 UAUDIT/NOAUDIT operand 29
 using when auditing 29
APPC/MVS
 activating auditing 24
 auditing considerations
 relationship of audit records 24
 transaction program auditing 24
 user verification requests 23
 deactivating auditing 25
 listing audit records 25
 persistent verification 23
APPLAUDIT 24, 28
APPLAUDIT key 140, 141
APPLAUDIT operand
 SETROPTS command 23, 25
APPLAUDIT suboperand
 APPLAUDIT suboperand of REASON
 operand 134
ASCEND operand 141
assistive technologies 191
ASSOCIATED GROUP column heading
 in the RACF started procedures table
 reports 110

ASSOCIATED USER column heading
 in the RACF started procedures table
 reports 110
ASSOCIATION TYPE association
 list of users with 113
associations
 ASSOCIATION TYPE 113
 NODE.USERID 113
 PASSWORD SYNC 113
ASSOCIATIONS column heading
 in the selected user attribute
 report 113
AT keyword
 auditing 39
 LOGSTR relocate section 39
attribute
 AUDITOR 1, 113
 group AUDITOR 1
 group-OPERATIONS
 logging activities for 19
 group-SPECIAL
 bypassing the logging of activities
 for 19
 logging activities for 19
 OPERATIONS 113
 logging activities for 19
 REVOKE 113
 SPECIAL 1, 113
 bypassing the logging of activities
 for 19
 logging activities for 19
ATTRIBUTE TYPE column heading
 in the selected user attribute
 report 113
audit controls
 general audit controls 17
 RACF/DB2 external security
 module 47
 setting 17
 system-wide 17
AUDIT operand
 SETROPTS command 18
audit tools 1
 audit control functions
 RACF SMF data unload utility 2
 DSMON 2
 logging routines 2
 RACF report writer 2
auditing
 access attempts 21, 22, 28
 access checks 28
 APPC transactions 23, 25, 28
 AT keyword 39
 checking DB2 authorization 44
 directory searches 28
 dubbing and undubbing of
 processes 28
 file system objects 28
 ONLYAT keyword 40
 RACF remote sharing facility 38, 39,
 40, 41, 42

auditing (*continued*)
 RACF/DB2 external security
 module 32
 RRSFDATA class 39, 41, 42
 SECLABEL profiles 28
 z/OS UNIX events 32
AUDITING column heading
 in the RACF class descriptor table
 report 105
auditor
 asking security questions 9
 audit tools provided 1
 auditor-controlled logging 4
 concept of accountability 1
 controlling auditing 3
 general audit controls
 AUDIT/NOAUDIT 18
 CMDVIOL/NOCMDVIOL 18
 LIST 18
 LOGOPTIONS 18
 OPERAUDIT/
 NOOPERAUDIT 18
 REFRESH GENERIC 18
 SAUDIT/NOSAUDIT 18
 SECLABELAUDIT/
 NOSECLABELAUDIT 18
 SECLEVELAUDIT/
 NOSECLEVELAUDIT 18
 group-wide auditor responsibilities 1
 listing specific audit controls 31
 logging specific events 3
 obtaining reports 119
 overriding owner-controlled logging
 specifying audit controls 4
 using the RACROUTE
 Request=AUTH exit routine 5
 responsibilities 1
 setting 17
 specific audit controls 28
 specifying data set controls 30
 system-wide auditor
 responsibilities 1
 use of DSMON 95
 use of the warning indicator 144
 using RACF report writer 9, 119
 verifying owner controlled logging 4
AUDITOR attribute 1
 auditing RACF system 1
 AUDITOR suboperand of
 AUTHORITY operand 134
 AUDITOR suboperand of REASON
 operand 134
 controlling auditing 17
 general audit controls
 AUDIT/NOAUDIT 18
 CMDVIOL/NOCMDVIOL 18
 LIST 18
 LOGOPTIONS 18
 REFRESH GENERIC 18
 SAUDIT/NOSAUDIT 18

- AUDITOR attribute (*continued*)
 - general audit controls (*continued*)
 - SECLABELAUDIT/
NOSECLABELAUDIT 18
 - listing users with 113
 - needed to run DSMON 95
 - specifying audit controls 29
- AUTH requests 5
- AUTHORITY operand
 - AUDITOR suboperand 134
 - BYPASSED suboperand 134
 - EXIT suboperand 134
 - FAILSOFT suboperand 134
 - NORMAL suboperand 134
 - OPERATIONS suboperand 134
 - SPECIAL suboperand 134
 - TRUSTED suboperand 134
- authorization 10
- authorization requests
 - auditing not done 21
- automatic command direction 40
- automatic password direction 42
- automatically directed application updates
 - auditing 41
- automatically directed commands
 - auditing 40
- automatically directed passwords
 - auditing 42

B

- BY(name2) operand 142
- BYPASS PASSWORD PROTECTION
 - column heading
 - in the program properties table report 103
- bypassing
 - logging of activities for SPECIAL attribute 19
 - logging of RACF command violations 20

C

- class descriptor table report
 - description 105
 - sample of 107
 - use of 105
- CLASS NAME column heading
 - in the RACF class descriptor table report 105
 - in the RACF global access checking table report 108
- CLASS operand 138
- classroom courses, RACF x
- command and subcommand processing
 - described 120
 - END subcommand 120
 - EVENT subcommand 120
 - LIST subcommand of RACFRW command 120
 - RACF report writer 120
 - RACRPORT EXEC 120
 - SELECT subcommand 120
 - SMFMERGE 120

- command and subcommand processing (*continued*)
 - SUMMARY subcommand 120
- command direction 39
- command syntax description 128
- command violations
 - auditing 27
- command-summary report 146
- commands
 - for RRSF 38
 - operator 38
 - RESTART 39
 - SET 38
 - STOP 38
 - TARGET 38
- COMPATMODE suboperand
 - COMPATMODE suboperand of REASON operand 134
- CONTROL universal access authority
 - in the selected data set report 117
- control user access 1
- controlling auditing 2
- controlling logging
 - by auditor 4
 - by owner 4
- courses about RACF x
- CPU MODEL column heading
 - in the system report 101
- CPU-ID column heading
 - in the system report 101

D

- DATA operand
 - RACFRW command 129
- data set controls 29
- DATA SET NAME column heading
 - in the selected data set report 115
- data sets
 - specifying control 30
- database, DB2
 - for IRRADU00 output 59
- DATASET class
 - auditing for 18
- DATASET operand
 - RACFRW command 129
- DATE operand 132
- DB2
 - creating a DB2 table space for IRRADU00 output 59
 - creating DB2 tables for IRRADU00 output 60
 - creating optimization statistics for the DB2 database 63
 - database for IRRADU00 output 59
 - DB2 utility statements required to delete the group records 63
 - deleting the IRRADU00 data from the DB2 database 63
 - loading the DB2 tables for IRRADU00 output 61
 - reorganizing the unloaded RACF SMF data in the DB2 database 63
 - SQL utility statements creating a table for IRRADU00 output 61
 - SQL utility statements defining a table space for IRRADU00 output 59

- DB2 (*continued*)
 - table names provided in SYS1.SAMPLIB 63
 - using with IRRADU00 output 58
 - utility statements required to load the tables with IRRADU00 output 62
- DB2 load utility
 - sample statements for IRRADU00 output 58
- DEFAULT UACC column heading
 - in the RACF class descriptor table report 105
- DEFINE request
 - logging activity for specified classes 18
- DESCEND operand 141
- DFSORT
 - ICETOOL 8, 54
 - customizing reports 57
- directed commands
 - AT keyword 39
 - auditing 39
 - ONLYAT keyword 40
- DSMON (data security monitor) 95, 107
 - class descriptor table report 105
 - control statements
 - LINECOUNT 97
 - USEROPT 97
 - control statements for DSMON functions 96
 - description 95
 - functions for generating reports 98
 - generated by DSMON
 - class descriptor table report 105
 - group tree report 102
 - started procedures table report 109
 - global access checking table report 108
 - group tree report 102, 109
 - how to run 95
 - JCL to run on MVS 95
 - list of reports produced 100
 - program properties table report 103
 - RACF authorized caller table report 104
 - RACF exits report 107
 - selected data sets report 115
 - selected user attribute report 112
 - selected user attribute summary report 114
 - shared RACF database 95
 - system report 100
- DSNAME operand
 - RACFRW command 129
- DSQUAL operand 138

E

- END subcommand of RACFRW command
 - description 143
 - syntax 143
- ENTRY NAME column heading
 - in the RACF global access checking table report 108

EVENT subcommand of RACFRW command
 ALLOWED operand 139
 CLASS operand 138
 dependency of SELECT subcommand 136
 described 136
 DSQUAL operand 138
 event-name operand 137
 event-name operand values listed 138
 EVQUAL operand 138
 INTENT operand 138
 issued with SELECT subcommand 131
 LEVEL operand 139
 monitoring
 access violations 146, 148
 NAME operand 138
 NEWSQUAL operand 139
 NEWNAME operand 139
 password violation levels 144
 syntax 136
 event-name operand 137
 events
 logging 3
 EVQUAL operand 138
 example 28
 DSMON reports 102
 EXECUTE access authority needed to run DSMON 95
 EXIT MODULE NAME column heading in the RACF exits report 108

F

failsoft processing
 logging 4
 FORMAT operand
 RACFRW command 129
 FUNCTION control statement
 control statements
 FUNCTION 97
 DSMON (data security monitor) 97
 functions
 that DSMON uses to generate reports 98

G

general audit controls
 how to use 17
 general resource class
 auditing for 18
 general resource controls 30
 GENSUM operand
 RACFRW command 130
 global access checking
 auditing not done 21, 22
 global access checking table report
 description of 108
 use of 108
 GLOBALAUDIT operand
 possible values to specify 30
 specifying data set controls 30
 group AUDITOR attribute 1

group AUDITOR attribute (*continued*)
 list of users with 113
 responsibility limits defined 1
 GROUP column heading
 in the group tree report 102
 GROUP operand 133
 group OPERATIONS attribute
 list of users with 113
 logging activities for 19
 monitoring
 group OPERATIONS users 147
 group REVOKE attribute
 list of users with 113
 group tree report
 description 102
 sample of 102
 use of 102
 group-AUDITOR attribute
 general audit controls
 LIST 18
 REFRESH GENERIC 18
 restriction for controlling auditing 17
 specifying audit controls 29
 group-SPECIAL attribute
 bypassing the logging of activities for 19
 list of users with 113
 logging activities for 17, 19
 monitoring
 group-SPECIAL users 147
 groups 31

I

ICETOOL
 generating reports 55
 IRRADU00 reports 56
 RACFICE PROC 55
 record selection criteria 55
 report
 example 56
 report format 54
 ICHRSMFE exit routine 119
 ICHRSMFI module 119
 SORTEQU field 140
 installation access control 1
 installation accountability 1
 installation exit ICHRSMFE 119
 installation-replaceable module
 ICHRSMFI 119
 INTENT operand 138
 Interactive System Productivity Facility (ISPF) program 2
 IRRADU00 utility
 creating a DB2 database 59
 creating a DB2 table space 59
 creating optimization statistics for the DB2 database 63
 creating the DB2 tables 60
 DB2 table names 63
 DB2 utility statements required to delete the group records 63
 DB2 utility statements required to load the tables 62
 deleting data from the DB2 database 63
 description 49

IRRADU00 utility (*continued*)
 effectively using 53
 example 51
 loading the DB2 tables 61
 operational considerations 49
 output 52
 reorganizing the unloaded RACF SMF data in the DB2 database 63
 SQL utility statements creating a table 61
 SQL utility statements defining a table space 59
 steps for using IRRADU00 output with DB2 58
 using 49
 using output with DB2 58
 IRRADULD
 member in SYS1.SAMPLIB 58
 IRRADUTB
 member in SYS1.SAMPLIB 58
 IRRDBU00 utility
 auditor's use of 7, 8
 IRRICE
 member in SYS1.SAMPLIB 8, 54
 SMF records 8
 IRRUT100 utility
 auditor's use of 7
 ISPF panels
 advantages 6
 compared to RACF TSO commands 6
 sample for password rules 7
 ISPF program 2

J

JCL (job control language)
 for DSMON 95
 JOB operand 133

K

key 24
 keyboard
 navigation 191
 PF keys 191
 shortcut keys 191

L

LDIRECT command 31
 LEVEL column heading
 in the group tree report 102
 LEVEL operand 139
 LFILE command 31
 LINECNT operand
 RACFRW command 130
 LINECOUNT control statement
 DSMON (data security monitor) 97
 LIST request
 list of programs authorized to issue 104
 LIST subcommand of RACFRW command
 ASCEND operand 141
 DESCEND operand 141

- LIST subcommand of RACFRW
 - command (*continued*)
 - description 139
 - NEWPAGE operand 141
 - restrictions 139
 - SORT operand 140
 - syntax 140
 - TITLE operand 140
- LISTDSD command 4, 31
 - using when auditing 31
- LISTGRP command 4, 31
- listing specific audit controls
 - described 31
 - LDIRECT command 31
 - LFILE command 31
 - LISTDSD command 31
 - LISTGRP command 31
 - LISTUSER command 31
 - RLIST command 31
- LISTUSER command 4, 31
- logging
 - access attempts based on security
 - level 20
 - access levels 4
 - accesses to
 - resources at specific audit levels 5
 - resources within a certain
 - SECLABEL 5
 - specific data sets 5
 - specific general resources 5
 - activities for OPERATIONS
 - attribute 19
 - activities for SPECIAL attribute 19
 - activities of group-OPERATIONS
 - users 17
 - activities of group-SPECIAL users 17
 - activities of OPERATIONS users 17
 - activities of SPECIAL users 17
 - all RACF-related activities for
 - users 28
 - attempts to access
 - DASD data sets 28
 - general resources 28
 - resources protected by a security
 - label 28
 - attempts to access RACF-protected
 - resources 29
 - auditor-controlled logging 4
 - AUTH requests 5
 - bypassing for users with SPECIAL
 - attribute 19
 - changes to any RACF profile 5
 - command violations 17
 - deletions to profiles 29
 - events RACF always logs 4
 - events RACF never logs 4
 - failsoft processing 4
 - general audit controls 17
 - general resource information 30
 - LIST requests 104
 - obtaining printed reports 9
 - owner-controlled logging 4
 - profile changes 17
 - RACF commands
 - issued by group-SPECIAL user 5
 - issued by SPECIAL user 5
 - violations 5

- logging (*continued*)
 - RACF commands issued 29
 - RACF data in SMF records 9
 - RACF-related activities 146
 - RACF-related activities of specific
 - users 5
 - RACROUTE REQUEST=VERIFY
 - requests 3
 - setting audit controls 17
 - specific events 3
 - system-wide for RACF classes 18
 - system-wide RACF command
 - violations 20
 - types of accesses 4
 - use of RVARY command 3
 - use of SETROPTS command 3
 - user additions to profiles 29
 - user changes to profiles 29
 - VERIFY requests 104
- LOGOPTIONS operand
 - SETROPTS command 21
- LOGOPTIONS suboperand
 - LOGOPTIONS suboperand of
 - REASON operand 134
- LOGSTR
 - AT keyword 39
 - RACF/DB2 external security
 - module 46
 - relocate section 39
 - using data 46

M

- management control 16
- messages
 - program properties table report 103
 - RACF authorized caller table
 - report 105
 - RACF exits report 108
 - selected data sets report 118
 - selected user attribute report 114
 - system report 101
- miscellaneous security concerns 11
- modifying resource profile 4
- MODULE LENGTH column heading
 - in the RACF exits report 108
- MODULE NAME column heading
 - in the RACF authorized caller table
 - report 105
- monitoring
 - access attempts
 - by security label 148
 - by security level 147
 - in warning mode 144
 - access violations
 - UACC 145
 - with RACF report writer 145, 146
 - with RACFRW command 148
 - OPERATIONS users
 - with RACF report writer 147
 - with RACFRW command 147
 - password violation levels
 - LOGON process 143
 - password violation occurrences 143
 - password violation stabilization 143
 - SPECIAL users
 - with RACF report writer 147

- monitoring (*continued*)
 - SPECIAL users (*continued*)
 - with RACFRW command 147
 - specific users
 - with RACF report writer 146
 - with RACFRW command, by
 - specified user 147
 - use of RACF commands
 - with RACF report writer 146
 - with RACFRW command, by
 - specified user 146
 - with RACFRW command, by
 - users 146

N

- NAME operand
 - EVENT subcommand of RACFRW
 - command 138
- name1 operand
 - SUMMARY subcommand of RACFRW
 - command 142
- navigation
 - keyboard 191
- NEWDSQUAL operand
 - EVENT subcommand of RACFRW
 - command 139
- NEWNAME operand 139
- NEWPAGE operand
 - LIST subcommand of RACFRW
 - command 141
 - SUMMARY subcommand of RACFRW
 - command 142
- NOCMDVIOL operand
 - SETROPTS command 20
- NODE.USERID association
 - list of users with 113
- NOFORMAT operand
 - RACFRW command 129
- NOGENSUM operand
 - RACFRW command 130
- NOJOB operand
 - SELECT subcommand of RACFRW
 - command 133
- NONE access authority
 - in the selected data set report 118
- NOOWNER operand 133
- NOSAUDIT operand
 - SETROPTS command 19
- Notices 195
- NOUSER operand
 - SELECT subcommand of RACFRW
 - command 133

O

- ONLYAT keyword
 - auditing 40
- OPERATING SYSTEM/LEVEL column
 - heading
 - in the system report 101
- OPERATIONS attribute
 - list of users with 113
 - logging activities for 19
 - monitoring
 - OPERATIONS users 147

OPERATIONS attribute (*continued*)
 OPERATIONS suboperand of
 AUTHORITY operand 134
 OPERATIONS column heading
 in the RACF class descriptor table
 report 105
 OPERATIONS user
 auditing 27
 operator commands
 for RRSF 38
 RESTART 39
 SET 38
 STOP 38
 TARGET 38
 OPERAUDIT operand
 SETROPTS command 19
 overriding user specification 30
 OWNER operand 133
 owner-controlled logging
 done by resource owner 4
 listing specific audit controls 31
 overridden by auditor 4
 overriding user specification 30

P

panels
 using 2
 password rules
 ISPF panel for 7
 PASSWORD SYNC association
 list of users with 113
 password violation levels
 calculating percentages 144
 example 144
 monitoring 143
 printed reports
 from DSMON 95
 from the RACF report writer 9
 printing reports from DSMON 95
 printing reports from the report writer 9
 PRIVILEGED column heading
 in the RACF started procedures table
 reports 110
 PROCEDURE NAME column heading
 in the RACF started procedures table
 reports 110
 PROCESS operand
 SELECT subcommand of RACFRW
 command 133
 PROCESS records 121
 profiles
 refreshing 25
 PROGRAM NAME column heading
 in the program properties table
 report 103
 program properties table report
 description 103
 messages 103
 sample of 104
 use of 103
 protection plan 12

R

RACF
 access control 1
 commands
 SETROPTS command 17
 controlling auditing 2
 RACF auditor
 definition of 1
 RACF authorized caller table report
 description 104
 messages 105
 use of 104
 RACF commands
 ALTUSER command 29, 146
 bypassing logging of violations 20
 LDIRECT command 31
 LFILE command 31
 LISTDSD command 4, 31
 LISTGRP command 4, 31
 LISTUSER command 4, 31
 logging activity for specified
 classes 18
 RALTER command 31
 RLIST command 4, 31
 RVARY command 3
 SEARCH command 4
 SETROPTS command 3, 17
 monitoring access to resources
 with a security label 148
 monitoring access to resources
 with a security level 148
 monitoring SPECIAL users 147
 RACF database
 sample DDL statements 58
 RACF exits report
 description 107
 messages 108
 sample of 108
 use of 107
 RACF global access table report
 sample of 109
 RACF implementation/integrity 10, 12
 RACF INDICATED field
 in the selected data set report 116
 RACF PROTECTED field
 in the selected data set report 117
 RACF remote sharing facility (RRSF)
 auditing 38
 RACF report writer 119, 122
 command description 123
 compared to RACF commands 9
 default values 122
 examples 149
 installation exit ICHRSMFE 119
 installation-replaceable module
 ICHRSMFI 119
 merging RACF for z/VM SMF records
 with RACF for MVS SMF
 records 172
 monitoring
 access attempts by security
 label 148
 access attempts by security
 level 147
 access attempts in WARNING
 mode 144
 access violations 145

RACF report writer (*continued*)
 monitoring (*continued*)
 OPERATIONS users 147
 password violation levels 143
 RACF commands 146
 SPECIAL users 147
 specific users 146
 obtaining reports 119
 overview 120
 record selection criteria 131
 report generation 122
 report types
 access to RACF-protected
 resource 119
 descriptions of group activity 119
 descriptions of user activity 119
 summaries of resource use 119
 summaries of system use 119
 return codes from 127
 SMF record types 121
 SMF records 9
 subcommand description 123
 terminal monitor program 123
 TMP 123
 use of SMF records 122
 use of work data set 122
 using 9
 warning mode example 145
 year 2000 119
 RACF Report Writer 119
 RACF SMF data unload utility
 description 49
 effectively using 53
 operational considerations 49
 RACF started procedures table report
 sample of 111, 112
 RACF TSO commands
 compared to ISPF panels 5
 RACF/DB2 external security module
 auditing 32
 checking authorization 44
 profile checking 45
 setting audit controls 47
 using LOGSTR data 46
 RACFICE
 generating reports 55
 IRRADU00 reports 56
 using 55
 RACFRW command
 command syntax 123
 command syntax description 128
 DATA operand 129
 default maximum record length 122
 description 128
 DSNAME operand 129
 END subcommand 143
 EVENT subcommand 136
 FORMAT operand 129
 GENSUM operand 123, 130
 LINECNT operand 130
 LIST subcommand of RACFRW
 command 139
 monitoring
 access violations 146, 148
 OPERATIONS users 147
 RACF commands 146

- RACFRW command (*continued*)
 - monitoring (*continued*)
 - RACF commands issued by user 146
 - SPECIAL users 147
 - specific users 147
 - NOFORMAT operand 129
 - NOGENSUM operand 130
 - sample reports 152
 - SAVE operand 129
 - SELECT subcommand 130
 - subcommand description 123
 - SUMMARY subcommand 141
 - syntax 128
 - TITLE operand 129
 - using RACF report writer 9
 - warning mode example 145
 - RACINIT AUTHORIZED column heading
 - in the RACF authorized caller table report 105
 - RACLINK command
 - auditing 42, 43, 44
 - examples 43, 44
 - phases 43
 - RACLIST AUTHORIZED column heading
 - in the RACF authorized caller table report 105
 - RACROUTE REQUEST=VERIFY request 3
 - RALTER command
 - GLOBALAUDIT operand 30
 - overriding user specification 30
 - using when auditing 31
 - READ access authority
 - in the selected data set report 117
 - needed to run DSMON 95
 - REASON operand
 - AUDITOR suboperand 134
 - CLASS suboperand 134
 - CMDVIOL suboperand 134
 - COMMAND suboperand 134
 - RACINIT suboperand 134
 - RESOURCE suboperand 134
 - SPECIAL suboperand 134
 - USER suboperand 134
 - record selection
 - RACF report writer 121
 - records
 - PROCESS records 121
 - SMF records 121
 - STATUS records 121
 - REFRESH GENERIC operands
 - SETROPTS command 25
 - REFRESH RACLIST operands
 - SETROPTS command 26
 - refreshing in-storage generic profiles
 - example 27, 28
 - reports
 - access to RACF-protected resource 119
 - customizing 57
 - descriptions of group activity 119
 - descriptions of user activity 119
 - general summary 158
 - reports (*continued*)
 - generated by DSMON
 - program properties table report 103
 - RACF authorized caller table report 104
 - RACF exits report 107
 - selected data sets report 115
 - selected user attribute report 112
 - selected user attribute summary report 114
 - system report 100
 - list of summaries from the report writer 141
 - listing of status 158
 - produced by DSMON 100
 - RACFRW sample reports 152
 - standard header page for report writer 157
 - summaries of resource use 119
 - summaries of system use 119
 - summary
 - group activity 161
 - group activity by resource 165
 - owner activity 164
 - owner activity by resource 154, 172
 - RACF command activity 163
 - RACF command activity by group 171
 - RACF command activity by resource 172
 - RACF command activity by user 170
 - resource activity 162
 - resource activity by group 167
 - resource activity by resource 166
 - resource activity by security event 168
 - security event activity 164
 - security event activity by resource 169
 - user activity 160
 - user activity by resource 165
 - using ICETOOL 56, 57
 - using RACFICE 56
 - resource owners
 - controlling logging 4
 - responsibilities of auditors 1
 - return codes
 - from RACF report writer 127
 - REVOKE attribute
 - list of users with 113
 - RLIST command 4, 31
 - RRSF (RACF remote sharing facility)
 - auditing considerations 38
 - automatically directed application updates 41
 - automatically directed commands 40
 - automatically directed passwords 42
 - commands 38
 - directed commands 39
 - operator commands 38
 - RACLINK command 42
 - examples 43, 44
 - phases 43
 - RRSFDATA class 39, 41, 42
 - RRSFDATA class
 - controlling auditing 39, 41, 42
 - rules
 - establishing password syntax ISPF panel 7
 - RVARY command 3
- ## S
- sample
 - ISPF panel 7
 - sample reports
 - DSMON report
 - system report 102
 - DSMON reports
 - class descriptor table report 107
 - group tree report 102
 - from the report writer 152
 - SAUDIT operand
 - SETROPTS command 19
 - SAVE operand
 - RACFRW command 129
 - SEARCH command 4
 - listing profiles that have the warning indicator 145
 - SECAUDIT suboperand
 - SECAUDIT suboperand of REASON operand 134
 - SECLABEL auditing
 - related system overhead 23
 - using the SECLABELAUDIT function 23
 - SECLABELAUDIT function
 - for auditing security labels 23
 - related system overhead 23
 - SECLABELAUDIT operand
 - SETROPTS command 22
 - SECLABELAUDIT suboperand
 - SECLABELAUDIT suboperand of REASON operand 134
 - SECLEVELAUDIT operand
 - SETROPTS command 20
 - security administrator
 - monitoring
 - SPECIAL users 147
 - SPECIAL attribute 1
 - use of RACF commands 146
 - security label
 - auditing access attempts to resources with security labels 22
 - security level
 - auditing access attempts 27
 - logging access attempts based on 20
 - security topics for RACF
 - classroom courses x
 - SELECT subcommand of RACFRW command
 - AUTHORITY operand 134
 - DATE operand 132
 - dependency of EVENT subcommand 136
 - GROUP operand 133
 - issued with EVENT subcommand 131
 - JOB operand 133
 - monitoring
 - access violations 146, 148

SELECT subcommand of RACFRW command (*continued*)

- monitoring (*continued*)
 - OPERATIONS users 147
 - SPECIAL users 147
 - specific users 147
 - use of RACF commands 146
- NOJOB operand 133
- NOOWNER operand 133
- NOUSER operand 133
- OWNER operand 133
- password violation levels 144
- PROCESS operand 133
- RACFRW command 130
- REASON operand 134
- restrictions 131
- STATUS operand 133
- STEP operand 133
- SUCSESSES operand 132
- syntax 132
- SYSID operand 133
- TERMINAL operand 135
- USER operand 132
- VIOLATIONS operand 132
- warning mode example 145
- WARNINGS operand 132

selected data sets report

- description 115
- messages 118
- sample of 118
- use of 115

selected user attribute report

- description 112
- messages 114
- sample of 114

selected user attribute summary report

- description 114
- sample of 114

SELECTION CRITERION column heading

- in the selected data set report 115

sending comments to IBM xiii

SETROPTS command

- AUDIT/NOAUDIT operands 18
- CMDVIOL/NOCMDVIOL operands 18
- examples 27
- LIST operand
 - LOGOPTIONS 18
 - SECLABELAUDIT/NOSECLABELAUDIT 18
- logging use of 3
- monitoring
 - access to resources with a security label 148
 - access to resources with a security level 148
 - group-SPECIAL users 147
 - SPECIAL users 147
- operands for auditing 17
- REFRESH GENERIC operands 18, 25
- REFRESH RACLIST operand 26
- SAUDIT operand 147
- SAUDIT/NOSAUDIT operands 18
- SECLABELAUDIT 148
- SECLEVELAUDIT operand 148

shared RACF database between MVS and z/VM

- considerations when running DSMON 95

shortcut keys 191

SMF data unload utility

- see RACF SMF data unload utility 49

SMF data unload, output to XML 65

SMF record

- type 44 relocate section 40, 42

SMF records 9

- listing contents of 119
- merging RACF for z/VM SMF records with RACF for MVS SMF records 172
- PROCESS records 121
- RACF report writer 119
- STATUS records 121
- types
 - type 20 121
 - type 30 121
 - type 80 121
 - type 81 121
 - type 83 121
- used by RACF report writer 121

SMF-ID column heading

- in the system report 101

SORT operand 140

SPECIAL attribute 1

- bypassing the logging of activities for 19
- list of users with 113
- logging activities for 19
- monitoring
 - SPECIAL users 147
- SPECIAL suboperand of AUTHORITY operand 134
- SPECIAL suboperand of REASON operand 134

SPECIAL user

- auditing 27

specific audit controls

- all RACF-related activities for users 28
- attempts to access
 - DASD data sets 28
 - general resources 28
 - resources protected by a security label 28
- specifying audit controls 4

standard header page for report writer 157

started procedures table report

- description 109
- use of 109

STATISTICS column heading

- in the RACF class descriptor table report 105

STATUS column heading

- in the RACF class descriptor table report 105

STATUS operand 133

STATUS records 121

STEP operand 133

SUCSESSES operand 132

SUCSESSES operand (*continued*)

- SUMMARY subcommand of RACFRW command 142

Summary of changes xv

summary reports from the report writer 141

SUMMARY subcommand of RACFRW command

- BY(name2) operand 142
- description 141
- monitoring
 - use of RACF commands 146
- name1 operand 142
- NEWPAGE operand 142
- SUCSESSES operand 142
- syntax 142
- TITLE operand 142
- used with EVENT subcommand 141
- used with SELECT subcommand 141
- VIOLATIONS operand 142
- WARNINGS operand 142

SYS1.SAMPLIB data set

- IRRADULD 58
- IRRADUTB 58
- IRRICE 8, 54
- sample DB2 mappings 53
- sample DFSORT mappings 53

SYSID operand 133

system information 10

SYSTEM KEY column heading

- in the program properties table report 103

system report

- description 100
- messages 101
- produced when RACF is inactive 95
- sample report 102
- using 100

SYSTEM RESIDENCE VOLUME column heading

- in the system report 101

system-wide auditor responsibilities 1

T

table space for DB2

- creating 59

technical security concerns 14

terminal monitor program 123

TITLE operand 140

- RACFRW command 129
- SUMMARY subcommand of RACFRW command 142

TMP (terminal monitor program) 123

TOTAL DEFINED USERS column heading

- in the selected attribute summary report 114

TOTAL SELECTED ATTRIBUTE USERS column heading

- in the selected attribute summary report 114

TRACE column heading

- in the RACF started procedures table reports 110

trademarks 197

- TRUSTED column heading
 - in the RACF started procedures table reports 110
- type 20 SMF record 121
- type 30 SMF record 121
- type 80 SMF record 121
- type 81 SMF record 121
- type 83 SMF record 121

U

- UACC (universal access authority)
 - monitoring the use of 145
- UACC field
 - in the selected data set report 117
- UPDATE universal access authority
 - in the selected data set report 117
- usage of attributes 13
- user
 - auditing 27
- user attribute 1
- user controls
 - data set controls 29
 - general resource controls 30
 - listing specific audit controls 31
 - using the ALTUSER command 29
- user interface
 - ISPF 191
 - TSO/E 191
- USER operand 132
- user responsibilities 1
 - auditor's role (user with AUDITOR attribute) 1
 - security administrator's role (user with SPECIAL attribute) 1
- user-controlled logging
 - done by resource owner 4
 - overridden by auditor 4
- user-written exit routine ICHRSMFE 119
- user-written module ICHRSMFI 119
- USERID column heading
 - in the selected user attribute report 113
- USEROPT control statement
 - DSMON (data security monitor) 97
- USRDSN field
 - in the selected data set report 116

V

- VERIFY request
 - list of programs authorized to issue 104
- verify user access 1
- VIOLATIONS operand 132, 142
- VMAUDIT suboperand
 - VMAUDIT suboperand of REASON operand 134
- VOLUME SERIAL column heading
 - in the selected data set report 115

W

- warning indicator 144
- warning mode
 - cautions when using 144

- warning mode (*continued*)
 - defined 144
 - monitoring access attempts 144
 - report example 145
 - warning indicator 144
- WARNING operand
 - SEARCH command 145
- WARNINGS operand
 - SELECT subcommand of RACFRW command 132
 - SUMMARY subcommand of RACFRW command 142
- work data set 122

X

- XML, obtaining output from SMF data
 - unload 65

Y

- year 2000
 - RACF report writer 119

Z

- z/OS UNIX
 - auditable events 36
 - auditing considerations 32
 - controlling auditing
 - classes 33
- z/OS UNIX System Services
 - auditing options 36, 37
 - commands 36
 - superuser authority 37



Product Number: 5650-ZOS

Printed in USA

SA23-2290-00

