RACF Starter System for MVS

Document Number GG24-3120-01

September 1992

International Technical Support Center
Poughkeepsie

Take Note!

Before using this information and the product it supports, be sure to read the general information under "Special Notices" on page xv.

Second Edition (September 1992)

This edition applies to Version 1 Release 9 of the RACF (Resource Access Control Facility) Program Product (5740-XXH) for use with MVS.

Order publications through your IBM representative or the IBM branch office serving your locality. Publications are not stocked at the address given below.

An ITSC Technical Bulletin Evaluation Form for readers′ feedback appears facing Chapter 1. If this form has been removed, comments may be addressed to:

IBM Corporation, International Technical Support Center
Dept. H52, Building 930
P.O. Box 950
Poughkeepsie, New York 12602-0950

When you send information to IBM, you grant IBM a non-exclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

Abstract

This document describes the process of installing and customizing a RACF Version 1 Release 9 starter system on MVS. It is intended to supplement the documentation supplied in the CBIPO package.

This document is intended for systems engineers, system programmers, security administrators and auditors who have to plan, install, implement, and audit RACF. It contains recommendations and examples on how to use and implement the functions as well as a short description of how they work.

A basic knowledge of RACF and MVS is assumed.

LS (183 pages)

Contents

Figures

Tables

Special Notices

This publication is intended to help IBM System Engineers and customers install and implement RACF Version 1 Release 9 (RACF 1.9). The information in this publication is not intended as the specification of any programming interfaces that are provided by RACF 1.9 for use by customers in writing programs that request or receive its services. See the Publications section of the IBM Programming Announcement for RACF 1.9 for more information about what publications are considered to be product documentation.

References in this publication to IBM products, programs or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only IBM′s product, program, or service may be used. Any functionally equivalent program that does not infringe any of IBM′s intellectual property rights may be used instead of the IBM product, program or service.

Information in this book was developed in conjunction with use of the equipment specified, and is limited in application to those specific hardware and software products and levels.

IBM may have patents or pending patent applications covering subject matter 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 the IBM Director of Commercial Relations, IBM Corporation, Purchase, NY 10577.

The information contained in this document has not been submitted to any formal IBM test and is distributed AS IS. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer′s ability to evaluate and integrate them into the customer′s operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environments do so at their own risk.

Reference to PTF numbers that have not been released through the normal distribution process does not imply general availability. The purpose of including these reference numbers is to alert IBM customers to specific information relative to the implementation of the PTF when it becomes available to each customer according to the normal IBM PTF distribution process.

The following terms, which are denoted by an asterisk (*) in this publication, are trademarks of the International Business Machines Corporation in the United States and/or other countries:

CICS/ESA
DB2
DFSMS
Hiperspace
IBM
MVS/DFP
MVS/ESA
MVS/SP
MVS/XA
OPC
VTAM
RACF

Preface

This document is intended to help systems engineers to plan, install, implement, and audit RACF.

How This Document is Organized

The document is organized as follows:

- Chapter 1, “Installation”

  This chapter describes the installation steps that must be completed before you IPL the system with RACF.

- Chapter 2, “Starter Groups and Users Structure”

  This chapter describes a group structure implementation and how to define TSO users to RACF.

- Chapter 3, “Data Sets”

  This chapter discusses RACF protection of data sets.

- Chapter 4, “General Resources”

  This chapter discusses RACF protection of general resources, such as terminals, programs, and other resources. It also contains information about security in a JES environment.

- Chapter 5, “Options”

  This chapter discusses some of the system-wide options that the RACF security administrator can specify.

- Chapter 6, “Tables and Exits”

  This chapter describes some of the RACF tables and installation exits.

- Chapter 7, “Utilities”

  This chapter describes the RACF utilities to list and perform maintenance functions on the RACF database.

- Chapter 8, “Reports”

  This chapter describes the Data Security Monitor reports and the predefined RACF reports using the RACF Report Writer.

- Chapter 9, “RACF ISPF Panels”

  This chapter describes how to maintain RACF profiles using ISPF panels.

- Chapter 10, “MVS Integrity”

  This chapter discusses the MVS integrity mechanisms to protect data security.

Related Publications

The following publications are considered particularly suitable for a more detailed discussion of the topics covered in this document.

- *Resource Access Control Facility (RACF):*
  - *Auditor's Guide*, SC28-1342
  - *Command Language Reference*, SC28-0733
  - *Data Areas*, LY28-1830
  - *Diagnosis Guide*, LY28-1016
  - *General Information*, GC28-0722
  - *General User's Guide*, SC28-1341
  - *Macros and Interfaces*, SC28-1345
  - *Master Index*, GC28-1035
  - *Messages and Codes*, SC38-1014
  - *Migration and Planning*, GC23-3054
  - *Program Directory for MVS Installations*, GC28-1054
  - *Security Administrator's Guide*, SC28-1340
- *System Programming Library: RACF*, SC28-1343
- *TSO/E Version 2:*
  - *Command Reference*, SC28-1881
  - *Customization* , SC28-1872
  - *Primer*, GC28-1879
  - *System Programming Command Reference*, SC28-1878
  - *User's Guide*, SC28-1880
- *MVS/ESA:*
  - *System Programming Library: Initialization and Tuning*, GC28-1828
  - *Planning: Security* , GC28-1604
  - *System Modifications* , GC28-1636
- *MVS/ESA and RACF Version 1 Release 9 Security Implementation Guide*, GG24-3585
- *MVS Storage Management Library: Managing Data Sets*, SC26-4408
- *CICS - RACF Security Guide*, SC33-0749

Acknowledgements

The advisor for this project was:

Valerie White
International Technical Support Center, Poughkeepsie

The authors of this document are:

Silvino Mezzari Junior
IBM Brazil

Patricia Candido de Oliveira
IBM Brazil

This publication is the result of a residency conducted at the International Technical Support Center, Poughkeepsie.

Chapter 1. Installation

This chapter describes the actions you must take after you install RACF* but before you IPL your CBIPO system. Only the actions described in Section 1.7, “Converting RACF Database to Restructured Format” on page 9, have to be performed after the initial IPL. The Custom-Built Installation Process Offering (CBIPO) methodology is used in this book.

Portions of this chapter are included in *MVS CBIPO Dialog Install Guide*. Examples of all jobs referred in this chapter can be found in Appendix A, “Starter System JCL” on page 133 and in IPO1.INSTLIB.

Installation process is different for new RACF installations and those who have previous releases of RACF currently operating.

1.1 RACF Database

Whether you currently have RACF on your system or are ordering RACF for the first time, you must allocate, format, and catalog a new RACF database on your target system. You should also create a backup database that RACF will maintain as a duplicate of the primary.

As indicated in RACF documentation, RACF 1.9.0 is the last release supporting the non-restructured database (NRDS) format. Therefore, it is strongly recommended that new users create a restructured RACF database. Also, installations that have RACF installed are encouraged to convert to the restructured database format (RDS) at this time.

Figure 1 on page 2 shows a comparison between the restructured database and the non-restructured database. The new restructured RACF database provides more efficient management of information and a flexible base for further enhancements, as follows:

- The block size is 4096 bytes, which will better utilize storage devices.
- RACF still allocates space with 256-byte segments, so there are 16 segments in each 4-KB blocks (K=1024).
- Profile names can now be up to 246 bytes in length. Longer names may be needed for some profiles, such as JESSPOOL profiles.
- In a non-restructured database, a profile must occupy contiguous space. Profiles in a restructured database need not be contiguous. For example, in a user profile, the TSO segment and the DFP segment can be separate from the RACF base segment. Storage requirements are reduced because the entire profile does not need to be accessed when only a few fields are required.
- There are no CONNECT profiles in an RDS. The information that is kept in a CONNECT profile in a NRDS is kept in the USER profile in an RDS. This results in increased processing efficiency because the CONNECT and USER profile are commonly used together. Programs using ICHEINTY or RACROUTE REQUEST=EXTRACT can continue to refer to CONNECT profiles, even though the data has been moved.

- New utilities are provided for the new RDS. There is a utility that corresponds to each of the ICHUTxxx utilities. Refer to Chapter 7, "Utilities" on page 73 for more detail about the new RACF utilities and their functions.

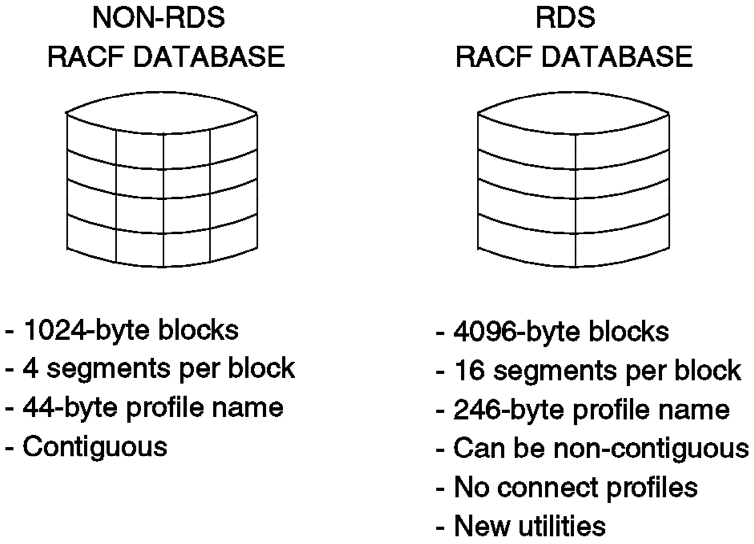


Figure 1. Restructured RACF Database

1.1.1 RACF Database Space Considerations

For performance reasons you may wish to change the SPACE allocation of the primary and backup databases in the SYSRACF DD statement in the allocation and initialization jobs from:

```
SPACE=(1024(900),,CONTIG)      -- NRDS
          or
SPACE=(4096(900),,CONTIG)      -- RDS
```

to:

```
SPACE=(CYL,(xx),,CONTIG)
```

Allocation in cylinders is better than allocation in block length, in terms of performance, because the RACF manager uses EXCPVR to access the RACF database.

There are two methods for estimating the number of cylinders required. The first is a detailed formula described in the *System Programming Library: RACF*. The second method is an approximation.

For both methods, the direct access space needed for a RACF database depends on the number of users, groups, user-group connections, and resource profiles defined to RACF, and the efficiency with which the space within the RACF database is utilized.

The algorithm used to estimate the RACF database size assumes that an installation uses only generic data set profiles. The algorithm uses the number of data set high level qualifiers (HLQs) in your system to estimate the database size. The number of HLQs is used because it includes the number of RACF users and RACF groups defined in your system. This number is then multiplied by an estimate of general resource profiles, user-group connections, and data set profiles that exist for each high level qualifier. We have used the number 50 after examining current installations. The RACF database requires approximately 320 bytes for each profile. The algorithm is:

No. of HLQs * 50 * 320 = Bytes required

Using Figure 2, which is derived from the formula above, an installation with a total of 1200 HLQs will need approximately 30 cylinders on a 3380 device. (This estimate includes some extra space for future expansion).

NO. OF	SPACE REQUIRED	3380 CYLS	3390 CYLS
HLQS	(BYTES)	(APPROX.)	(APPROX.)
200	3,200,000	6	5
400	6,400,000	10	9
800	12,800,000	20	17
1200	19,200,000	30	25
1600	25,600,000	40	34
2000	32,000,000	50	42
2400	38,400,000	60	50
2800	44,800,000	70	59
3200	51,200,000	80	67
3600	57,600,000	90	75
4000	64,000,000	100	83
5000	80,000,000	125	105

Figure 2. RACF Database Size Table

The RACF database must be cataloged and should reside on a permanently resident volume. Place the RACF database on the fastest available device for performance, even though it can be allocated on any DASD device available in your installation. If a cached auxiliary storage subsystem (such as IBM 3390 or 3380 devices attached to an IBM 3990 Storage Control Model 3) is available, use it for the RACF database. If the device is generated as a shared device, performance will be degraded, even though the RACF database is not actually being shared with another system.

1.1.2 RACF Database Allocation For New Installations

If this is the first time you have installed RACF on your MVS system, submit the job RACRDSP to allocate and initialize a primary RDS. You should also submit the job RACRDSB to allocate and initialize a backup RDS which RACF will maintain as a duplicate of the primary.

You may want to consider changing the SPACE allocation of the SYS1.RACF and SYS1.RACF.BKUP1 databases on the SYSRACF DD statement in jobs RACRDSP and RACRDSB. Refer to Section 1.1.1, "RACF Database Space Considerations" on page 2 for more information about space allocation.

1.1.3 RACF Database Allocation For Existing Installations

If you have previously installed RACF on your MVS system, submit jobs RACFINT and RACINBK to allocate and initialize primary and backup RACF NRDS.

You should change the SPACE allocation of the SYS1.RACF and SYS1.RACF.BKUP1 in jobs RACFINT and RACINBK. Refer to Section 1.1.1, "RACF Database Space Considerations" on page 2.

After the initial IPL, when the new RACF is active on the target system, the non-restructured databases can be converted to the restructured format.

1.1.4 Additional Consideration for Existing RACF Installations

After you have allocated a new RACF database and, optionally, a backup database, you can copy the SYS1.RACF database from the driving system to the new RACF database on the target system in order to retain your RACF profiles. Jobs RACFCPR and RACCPBK use RACF utility ICHUT200 to accomplish this task.

The RACF database must have templates at the 1.9 level if RACF 1.9 is to function properly. To upgrade the existing RACF database templates to the new release, use the new RACF initialization program, ICHMIN00. If your installation already uses restructured database, use program IRRMIN00.

Jobs RACUPGNP and RACUPGNB upgrade the target system′s primary and backup non-restructured databases, respectively.

Make sure you point to the RACF 1.9 libraries to pick up ICHMIN00. The RACF 1.8 copy of ICHMIN00 will update the templates correctly, but will not update new default values, and thus may cause problems after you IPL RACF 1.9. Also, in previous releases, the RACF Template Data (ICHTEMP0) resides in SYS1.MACLIB. In RACF 1.9, ICHTEMP0 resides in SYS1.MODGEN. Be sure to point to this library when running ICHMIN00.

You can run ICHMIN00 against an active RACF database. A reserve is issued for integrity purposes, but the elapsed time is short. A re-IPL is necessary for the template changes to take effect on the system. Note that the updated RACF databases can still be used by a RACF 1.8 or earlier system.

Note: After updating the database to RACF 1.9 templates, only the RACF 1.9 utilities may be used on the database, and they can only be run from a RACF 1.9 system. The only exceptions are ICHMIN00 and ICHUT100, which can be run from the RACF 1.8 system. Do not use RACF 1.8 or earlier utilities; otherwise, unpredictable errors can occur. You should take RACF 1.8 and earlier utilities out of the system load libraries.

Refer to Chapter 7, “Utilities” on page 73 for a list of the new RACF utilities and their counterparts in previous RACF releases.

Initialization program ICHMIN00 does not convert a non-restructured database (NRDS) to the restructured database (RDS). The installation must initially install RACF using the NRDS. After the initial IPL, when the new RACF is active on the target system, the NRDS can be converted to the restructured format. Refer to Section 1.7, “Converting RACF Database to Restructured Format” on page 9 for more information on the conversion to the RDS.

Conversion of the RACF database to the RDS format is not required in order to install RACF 1.9. However, RACF 1.9 is the only release in which the transition to the new format can be made.

1.2 Defining the RACF Database Name

During IPL, the system must have the name of each RACF database. If databases names are not defined prior to IPL, the operator is prompted for them. The operator can respond with the correct name or ′none′. If ′none′ is the reply, RACF will be inactive for this IPL and the operator will be prompted for all resources accesses.

To define the RACF database name automatically to the system and to eliminate operate intervention, either:

- use the database name table, ICHRDSNT or
- modify MSTRJCL.

The database name table is the recommended method. It provides the ability to identify a backup database for RACF′s use and specify what type of backup information is to be copied from the primary database to the backup database. It also provides better performance since you can specify resident data blocks.

1.2.1 Database Name Table

The Database Name table (ICHRDSNT) is an installation-written load module that describes the RACF databases to RACF. In this table you can specify the:

- Number of RACF databases
- Names of RACF databases (primary and backup)
- Number of resident data blocks
- What kind of information you want to write to the backup database

Figure 3 shows a sample database name table for one primary and one backup database.

```
DC   AL1(1)                    Number of entries
DC   CL44'SYS1.RACF'           Primary database name
DC   CL44'SYS1.RACF.BKUP1'     Backup database name
DC   AL1(255)                  Number of resident blocks
DC   B'10000001'               Flags - duplex updates
                               and resident blocks will
                               include data blocks
```

Figure 3. Database Name Table (ICHRDSNT)

The flag field has the following format:

Bit Setting Meaning

00.. .... No updates to backup database.

System problems may cause the backup database to become the primary database. In this case, the backup database should be reasonably accurate.

Use ICHUT200 or IRRUT200 utilities to make copies periodically of your primary database. For a sample of the necessary JCL, refer to jobs RACCOPY in Section A.9, “Copy Primary Non-restructured Database to Backup” on page 141 and RACRCPY in Section A.10, “Copy Primary Restructured Database to Backup” on page 142.

10.. .... Update backup database, except statistics.

The backup database changes only when the primary database changes.

In this case, if the SETROPTS INITSTATS is active, a limited subset of statistics is maintained on the backup database, such as the first time each day that the user logs on and the time and date of password changes during logon. Also, if SETROPTS PASSWORD(REVOKE) option is active, the revoke counter is reset to zero.

11.. .... Update backup database, including statistics.

Duplicating updates and statistics would increase the level of activity in the RACF backup database. You normally would not choose this option.

....1 Use the resident data-block option for the primary database. (Resident data blocks are always used on a restructured database.)

The resident index-block count field specifies the maximum number of index blocks to be kept resident for the primary database while it is active. If you select the resident data block option (setting bit 7 in the flag field), this field specifies the maximum number of index, block-availability-mask (BAM), and profiles blocks to be kept resident.

Resident data-blocks reduce the amount of I/O that is required to service the RACF database. Their use is highly recommended with the non-restructured database for best performance. With a restructured database, the resident data block option is defaulted. It is therefore unnecessary to set bit 7.

1.2.2 Recommended Number of Resident Blocks

Resident data blocks reduce the amount of I/O that is required to the RACF database. An installation can specify from 0 to 255 resident blocks per primary RACF database. For MVS, the default value is 10 resident blocks. For best performance, specify a number as large as you can afford.

In a NRDS, each data block uses 1056 (1 KB + 32) bytes of storage. In a RDS, each data block uses 4128 (4 KB + 32) bytes of storage. On MVS, the storage is in the CSA (extended CSA for MVS/XA* and MVS/ESA*).

Since the blocking factor for restructured data sets is four times greater than for non-restructured data sets, the CSA increase in like manner.

If you are in an MVS/370 environment, check the CSA size. If you have enough CSA, you should specify between 40 and 50 resident blocks.

If you are in an MVS/XA or MVS/ESA environment, you should specify 255 resident blocks. You do not have to worry about CSA size, because, in those systems the residents blocks are placed above the 16-MB line (M=1,048,576).

You can specify the number of resident blocks in ICHRDSNT or in ICHSECOP (when you do not have a database name table). RACF establishes this value in the following sequence:

1. Searches the database name table (ICHRDSNT)
2. Searches ICHSECOP
3. Uses the default value (10) from ICHSECOP

We recommend that you specify the number of resident blocks in ICHRDSNT.

For sample JCL to install the RACF Database Name table using SMP/E, refer to job RACDSN in A.11, “Install RACF Database Name Table” on page 143.

1.2.3 Modifying MSTRJCL

Even though it is recommended you use the database name table to define the database name to the system, you can also do this definition by modifying the MSTRJCL.

Add the following assembler source to the MSTRJCL CSECT using a JCL DD statement labeled SYSRACF:

```
  DC  CL80'//SYSRACF   DD   DSN=SYS1.RACF,DISP=SHR'
```

Link-edit the MSTRJCL object code into SYS1.LINKLIB.

MSTRJCL cannot be used to allocate a backup data base. For more information on updating MSTRJCL, see *MVS/ESA System Modifications*.

1.3 Data Encryption Standard Modification

The Data Encryption Standard (DES) is an algorithm that RACF uses to encode passwords and Operator Identification Card (OIDCARD) data before they are stored in the RACF database. It is a one-way encryption; it does not provide the reverse process (decryption). DES provides a higher level of security than the masking algorithm offered with RACF releases earlier than RACF 1.6.

For compatibility with prior releases, RACF provides a dummy authentication installation exit (ICHDEX01). With this exit, RACF uses only the masking algorithm. If you want to use an encrypted RACF password, you have to delete ICHDEX01 in SYS1.LPALIB to activate the DES algorithm.

If you previously installed RACF on shared systems and used DES, you must be certain that ICHDEX01 is deleted before IPLing your system. Otherwise, RACF will use the masking algorithm to read the passwords, and no user will be able to log on to your system. Module ICHDEX01 will be reinstalled and it must be deleted again. Delete this module for each system that shares the database and specify the CLPA option when you IPL your systems.

If you want to use DES for the first time, or you have previously used DES, run job RACDES. This job does two things:

- Deletes module ICHDEX01 from SYS1.LPALIB
- Changes the SMP/E owner of ICHDEX01 from the RACF FMID to IPOSDX0.

With this job, you can avoid the reinstallation (and re-deletion) of this module at each new RACF release and SYSGEN. It changes the owner of ICHDEX01 for a dummy FMID. Refer to Section A.12, “DES Modification” on page 145 for an example of job RACDES.

1.4 TSO/E APF-Authorized Command and Program Tables

The TSO/E APF-authorized command and program tables must contain the RACF commands if RACF is to operate properly.

If you have TSO/E Release 1.4 or higher on your system, use the SYS1.PARMLIB member IKJTSO00 to define the APF-authorized RACF commands and programs.

If you have installed RACF through CBIPO, IKJTSO00 contains the appropriate RACF commands and programs. You should verify that this member meets your requirements. You may modify IKJTSO00 at any time.

1.5 Placing Modules in LPA

In an IMS or CICS environment, you can place some RACF load modules in the fixed Link Pack Area (LPA) to ensure optimum performance. Put the modules in LPA by creating or modifying your existing IEAFIXxx list member in SYS1.PARMLIB.

If your installation wants to use RACF for IMS and CICS security functions, update IEAFIXxx with the following information:

```
SYS1.LPALIB     ICHRFC00,ICHRFR00,ICHSFR00
```

Users of IMS should also add the following to member IEAFIXxx:

IMS Release 2.1 or earlier:

```
SYS1.LINKLIB    ICHRGL00,ICHRGL01,ICHRGL03
```

IMS Release 2.2 or later:

```
SYS1.LINKLIB    ICHRGL03,ICHRGL04
```

Starting with CICS/ESA* Version 3 Release 2, security functions for a CICS environment must be provided by a security program, such as RACF. Internal security mechanisms to control the access to CICS resources are not provided by CICS anymore.

An example of IEAFIXxx can be found in SYS1.SAMPLIB member RACINSTL. See *System Programming Library: RACF* and *MVS/ESA System Programming Library: Initialization and Tuning* for detailed information.

1.6 Utilizing VLF for Group Tree in Storage

On an MVS/SP 3.1.0 or later system, RACF can improve the performance of group authority processing for RACF commands. This facility improves the performance of group authorities such as Group-special, for users with a large number of group connections.

RACF uses virtual lookaside facility (VLF) to create group information objects in data spaces, thus reducing RACF database I/O during group authority processing.

Use member COFVLFxx in SYS1.PARMLIB to activate or deactivate this Group Tree in Storage function, by including or omitting the IRRGTS class from COFVLFxx.

Group Tree in Storage must not be activated unless all RACF system sharing the RACF database are using RACF 1.9 or later. Refer to *MVS/ESA System Programming Library: Initialization and Tuning* for more details on coding member COFVLFxx.

1.7 Converting RACF Database to Restructured Format

This section is applicable only for installations migrating from a prior release of RACF, or installations that have installed RACF initially with a non-restructured database.

After RACF 1.9 with the NRDS is running on your system, you may decide to convert your databases to the RDS format. The RACF database conversion utility program (IRRDSC00) converts your RACF databases to the new format.

1.7.1 Running a Test Conversion

You can use utility IRRDSC00 to convert an off-line copy of your NRDS database to RDS format. Create the off-line copy using either utility ICHUT200 or utility ICHUT400. This off-line copy of the RACF database should reside on a different volume than your active RACF database to permit you to run your test conversion without impacting the active system.

This test will show whether all your profiles can be read and converted, and whether your restructured databases should be larger than the non-restructured databases. Run utility IRRUT200 to compare the database sizes. You can also estimate how long the conversion will take.

Utility IRRDSC00 reads and writes every profile in the entire database. If you want the conversion to run faster and provide less contention with other work, it is recommended that you run it during off-shift hours.

1.7.2 Converting the Active Database

After you have run the test conversion, convert your active RACF NRDS to the new RDS format.

One method is to install a RDS as the primary, and keep an NRDS as a backup. This approach permits your installation to have the benefits of the RDS, while retaining a NRDS in case of problems, until you feel your system is stable. At that time, you can use RACF utility IRRUT200 or IRRUT400 to copy your primary RDS to a backup RDS.

Note that once you have converted your database, you may not return that database to the non-restructured format. Therefore, it is recommended that you follow the method described above.

Remember that you must use utility IRRUTxxx for restructured databases and ICHUTxxx for non-restructured databases. Be especially careful during mixed mode operation, when your primary database is in a different format than your backup database. Also, remember that RACF Release 9.0 is the last release that will support the NRDS format.

For example, assuming that your system has a primary NRDS named SYS1.RACF and a backup NRDS named SYS1.RACF.BACKUP, you can use the following steps to convert the primary database to the RDS format:

1. Run utility IRRDSC00 to read the primary data set and construct a new RDS (for example, SYS1.RACF.RDB). Use the parameter LOCKINPUT to prevent updates while the job is running.
2. Use the RVARY INACT command to inactivate the backup data set SYS1.RACF.BACKUP.
3. Create a backup copy of the backup data set, to be used in case of problems during the conversion. If the conversion is successful, this data set can be deleted. Use ICHUT200 or some other utility to make the copy.
4. Copy the primary data set SYS1.RACF to the backup data set SYS1.RACF.BACKUP.
5. Use the RVARY ACTIVE command to activate SYS1.RACF.BACKUP.
6. Use the RVARY SWITCH command to make SYS1.RACF.BACKUP your primary data set and SYS1.RACF the backup data set.
7. Rename data set SYS1.RACF (the current backup data set) to, for example, SYS1.RACF.OLD.
8. Rename the output data set of IRRDSC00, SYS1.RACF.RDB (step 1), to SYS1.RACF.

At this time, your current backup data set is in the RDS format, and has the same name (SYS1.RACF) that the primary data set had at the beginning of the process.

9. Use the RVARY ACTIVE command to activate backup data set SYS1.RACF.

10. Use the RVARY SWITCH command to switch data sets SYS1.RACF and SYS1.RACF.BACKUP.

    Now SYS1.RACF is the restructured primary data set, and SYS1.RACF.BACKUP is the non-restructured backup data set.

    When you feel your system is stable, you can copy the primary RDS to a backup RDS.

Refer to *MVS/ESA and RACF Version 1 Release 9 Security Implementation Guide* for a complete discussion of the above method. *System Programming Library: RACF* contains additional details and sample JCL to perform the conversion to the RDS format.

Chapter 2. Starter Groups and Users Structure

Every new installation has to create a RACF groups and users structure. This structure, determined by the installation′s security needs, influences RACF administration. This chapter discusses ways to develop and administer a RACF groups and users structure.

Examples of the jobs referred in this chapter can be found in Appendix A, "Starter System JCL" on page 133 and in IPO.JCLLIB.

2.1 Group and User Concepts

A basic RACF concept is to distinguish between actual resources, such as data sets and tape volumes, and the RACF profiles that protect these resources. The RACF profiles are grouped into functional classes, called resource classes. For example, data sets are grouped in the DATASET class, and user IDs are grouped into the USER class. There is also a GROUP class that contains collections of users.

A RACF user is identified by an alphanumeric user ID that RACF associates with the user. Note, however, that a RACF user need not be an individual. For example, a user ID can be associated with a started task.

Users with similar access requirements can be collected in a group. This is very useful for users that work on the same project or in the same area and must have access to the same resources. Instead of authorizing each user to access the resource, you authorize the group. The group is easier to control, and the size of the database decreases because the list of users that access profiles is smaller.

2.1.1 Group Characteristics

The most important characteristics of groups are :

- Groups are organized hierarchically. This means that every group has a superior group (with the exception of the group SYS1, which is the major group).
- The high-level qualifier of a data set profile must always be a group (or a user) defined to RACF.
- Scopes of groups can be created. A user with the Group-special attribute in a group controls the profiles in a subgroup, if the profile owner is the subgroup and the superior group owns the subgroup profile.
- RACF profiles are identified by a profile name and contain a list of all users and groups that are authorized to access the resource related to these profiles (the access list).
- RACF can be administered at the system or group level.

2.1.2 Group Benefits

The benefits of using RACF groups include:

- Reducing the effort of maintaining access lists. Instead of adding and deleting users to the access list of several profiles, consider creating a RACF group and placing it on the access list instead of the user IDs.
- Avoiding the need to refresh in-storage profiles. If your installation maintains in-storage copies of resource profiles through the SETROPTS RACLIST or GENLIST, changes to those profiles do not take effect on the system until a SETROPTS RACLIST REFRESH or SETROPTS GENERIC REFRESH command is issue. To avoid the need to refresh in-storage copies, place the RACF group in the access list instead of a user ID. Refer to 5.6, “Using SETROPTS RACLIST and SETROPTS GENLIST” on page 57 for more information about RACLIST and GENLIST.
- Providing a form of time-limited access. First you have to ensure that the only access the user has to the resource is through a RACF group that has the desired access to the resource. Then, connect the user to the group with a resume or revoke date. To cause the user′s access to start on a certain date, issue the following command:

  ```
  CONNECT userid GROUP(groupid) RESUME(date)
  ```

  To cause the user′s access to stop on a certain date, issue the following command:

  ```
  CONNECT userid GROUP(groupid) REVOKE(date)
  ```

2.2 Skeleton Group Structure

This section describes how to create an initial RACF group structure.

2.2.1 General Considerations

Computer resources are always changing; data sets are created and deleted, and users are moved from one department to another. For this reason, the group structure should be easily changeable.

Whenever possible, access should be given to groups rather than individuals. This approach allows the system to be maintained more easily because the access list entries in the resource profiles are more stable. When a user changes groups, the resource access lists do not have to be changed. However, when only one user needs access to another group′s resources, only the one user should be permitted access to the resource, not the entire group.

How many groups and subgroups an installation decides to implement depends on the number of users and the complexity of the installation′s organization. At a smaller installation, one system programmer is sometimes responsible for several subsystems, so it would not be necessary to divide the system management group structure into groups for each separate subsystem. But keep in mind that installations grow and the installation should plan its group structure based on a reasonable growth plan.

Figure 4 shows the first level of the suggested starter group structure for all installations. The upper part of the figure shows the groups defined in the system when RACF was installed. The lower part of the figure describes the highest functional levels in a computer center: system management (SYSMGT), applications (APPLS), and end users (USERS). Logically the group SYSADM is established above these groups. This group contains users with the System-special and the System-auditor authority. In larger installations, it can be convenient to define more than one user with System-special or System-auditor authority.

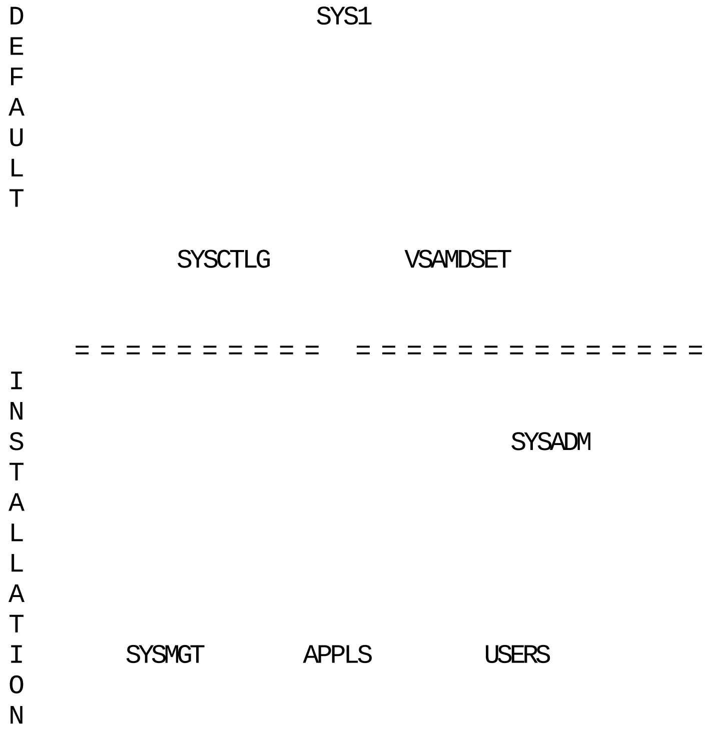


Figure 4. First-level Group Structure

2.2.2 System Management Group Structure

Many system data sets need the high-level qualifier SYS1. Therefore, with SYS1 data sets it is advantageous to create separate groups for data sets and users. For example, the group SYS1 would contain only data sets, and the groups shown in Figure 5 would contain only users and data sets used within the group, not data sets used system-wide. The main function of the defined groups is to collect users with similar access requirements. These groups are called access groups. Then, these access groups would be given access to the groups of SYS1 data sets associated with their particular system management function. In this figure, a functional structure is used. The group SWMGT contains the system programmers. This group covers all software maintenance and is split into subgroups responsible for the system software (SYSSW), such as JES, RACF, and TSO; the network software (NETSW), such as VTAM*; and the database software (DBSW), such as IMS, CICS, and DB2*. If the system programmer′s responsibility is more restricted, then this structure can be expanded to the product level.

In Figure 5, started procedures are defined in the group STCGRP. These procedures can access data sets in the group SYS1 only with explicit permission. The group STGMGT is responsible for the maintenance of storage, such as tapes and DASD volumes.

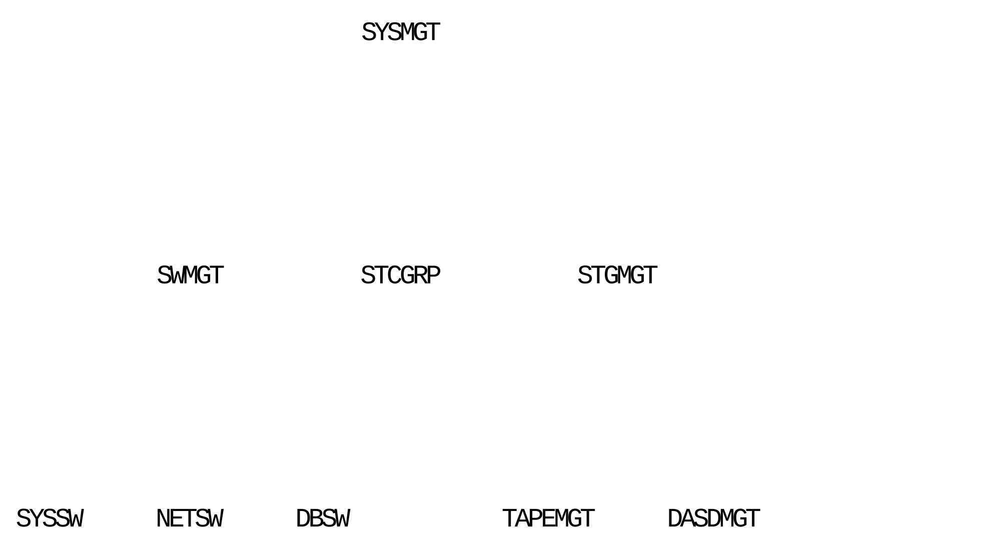


Figure 5. Substructure of System Management Group

For the JCL to define the system management group structure, refer to job RACGRP in Section A.13, “Skeleton Group Structure” on page 146. This job can be changed to suit the installation′s group structure requirements. The System-special attribute is required to execute this job.

Note: If Group Tree in Storage is not activated in your CBIPO system, MVS/XA and MVS/ESA (including Version 4) installations will receive the following message when RACGRP is run:

```
IRR804I RACF CLASS “IRRGTS” NOT DEFINED IN COFVLFXX PARMLIB of VLF.
```

This message will not affect your job. You may wish to refer to Section 1.6, “Utilizing VLF for Group Tree in Storage” on page 9 and to the *RACF Program Directory for MVS Installations* for information concerning Group Tree in Storage.

2.2.3 Applications Group Structure

Applications include dialogs, program products, and other programs used by several users. Applications exist on the group or system level. Group applications can be group-specific interfaces to subsystems, such as input panels for a finance package. If more than one group has to use the application, the application should be defined as a system application.

The structure of the applications groups should reflect the functional organization of the application. For example, programmers working on the application would be placed in one group and end users of the application would be placed in another group.

Often, applications are the interface between the computer center and the users. If the users believe that their daily work is restricted by RACF, the acceptance of security activities is low. Therefore, it is suggested that all departments be involved in the process of defining the RACF group structure.

2.2.4 User Group Structure

A RACF group structure should be based on the most stable part of an installation′s organization. For example, because department numbers are often changed, we recommend a user-group structure based on the functional organization of an installation. If temporary project groups are necessary, create them on the bottom of the hierarchical structure so that when they are deleted no other groups in the structure are affected.

2.2.5 Defining Groups

Use the ADDGROUP command to define one or more RACF groups. Be sure to specify at least your installation′s defined superior group and group owner, as follows:

```
ADDGROUP group SUPGROUP(supgroup) OWNER(group)
```

2.3 RACF Administration

RACF offers an installation a wide range of functions to help build and administer a security system around the RACF group structure. This system can be administered on the system level with centralized administration or on a group level with decentralized administration.

The installation must decide which method of security administration best suits its needs. For a new installation or a smaller installation, centralized administration is beneficial because this approach requires only RACF skills from one or two system administrators. Later, if the effort to administer RACF increases, the installation can implement a decentralized administration. For easy implementation of decentralized administration, the installation should initially define the scope of groups with group-level administration in mind.

2.3.1 Centralized Administration

With centralized administration, the system is administered by one or more system administrators defined with System-special authority. These administrators are be connected to the group SYSADM, discussed previously in Section 2.2.1, “General Considerations” on page 14. The system administrator controls all RACF profiles, permits users and groups to access the resources protected by these profiles, and defines or deletes users and groups as necessary throughout the entire group structure. This administrator also monitors the security requirements and implements protection measures for the entire installation.

2.3.2 Decentralized Administration

With decentralized administration, the security system is administered by delegates with Group-special authority at the group level and by a system administrator who administers only system-wide authorities. The group-level administrator controls all RACF profiles within the scope of his group, permits users and groups to access the resources protected by these profiles, and defines or deletes users and groups as necessary within the scope of his group.

A group-level administrator understands the needs of a group much better than a system administrator because he is closer in the organization to the group members and is more aware of the group′s specific security requirements. To guarantee system integrity, the group-level administrators are controlled by the system administrator. All administrators are monitored on the same level by users with the Auditor or Group-auditor attribute.

2.4 Group Structure Implementation

This section describes steps to be taken to customize RACF on the system after IPL. Consult the *RACF Command Language Reference* when defining the parameters for the commands to be certain that they accurately reflect your installation′s desired RACF implementation.

Implement your group structure as follows:

1. Initially, you should be logged on as user ID IBMUSER. This user ID is the only access to both RACF and TSO after installing RACF. Change the initial password of "SYS1" to ensure that only authorized users can access this user ID.

2. Define the SETROPTS options. See Chapter 5, "Options" on page 53.

3. Define your RACF group structure.

4. Define users with the Special attribute. The Special user who is the new security administrator must have a SYS1.UADS entry with TSO ACCOUNT authority and must have SYS1 as the default group. To create this Special user ID, issue the following RACF command from TSO:

   ```
   ADDUSER userid PASSWORD(password) DFLTGRP(SYS1) OWNER(SYS1) SPECIAL
   ```

5. Define users with Operations and Auditor attributes.

6. The security administrator then logs on to his new Special user ID created above. He will be prompted for a new password. User ID IBMUSER should be revoked to prevent its further use. From the security administrator′s user ID, issue the following RACF command from TSO:

   ```
   ALTUSER IBMUSER REVOKE
   ```

   All jobcards in the RACF sample jobs must be changed to reflect the user ID of the person executing the jobs, such as the security administrator.

7. Define all TSO user IDs to RACF.

8. Define all exclusively batch users to RACF.

9. Define high-level qualifiers that do not match a RACF user ID as a RACF group.

10. Protect all high-level qualifiers. To do so, create simple generic profiles (such as HLQ.*) in the RACF DATASET class for each user and group with the WARNING option. With the WARNING option, access requests are not denied. A user attempting the access receives a message from RACF and RACF SMF records are produced.

11. Customize Tables and Exits - Global Access Checking Table, Started Procedures table, RACF Installation Exits, and others. See Chapter 6, "Tables and Exits" on page 59.

12. Run the report writer to identify whether more permits are necessary. The Report Writer shows all attempts to access profiles in Warning mode. When

you turn off the warning mode the user cannot access the profile, so you have to analyze these accesses to see whether users must be included in the access list of the profile or whether it is necessary to create more precise generic profiles to protect the data sets. Section 8.1, “Report Writer” on page 81 contains the description of some RACFRW predefined reports.

13. Run DSMON to check the initial steps you took to establish system security. See Section 8.2, “Data Security Monitor” on page 86 for information about the use of DSMON.
14. When you are sure that your system is running with no problems, reset the warning mode of the profiles.

2.5 Defining TSO Users to RACF

To log on to TSO/E, a user must have an entry in the SYS1.UADS data set or a TSO segment defined in the RACF user profile.

On an MVS/XA or MVS/ESA system, you can move TSO/E user attribute information from SYS1.UADS to the RACF database. When you move this TSO/E information into the RACF database, it is stored in the TSO segment of the user′s profile. When a user logs on to TSO/E, TSO/E uses the information contained in the RACF TSO segment to build a session for the user.

We strongly recommend that users be converted from SYS1.UADS to the RACF database to improve the security of your systems and to simplify the maintenance of user ID information.

To convert user information from the UADS to the RACF database, follow these steps:

1. Convert user information using the TSO/E RACONVRT command.
2. Test the system using the TSO information in the RACF database.
3. Delete user information from the UADS.
4. Synchronize the RACF database with the broadcast data set.

The following sections describes the above steps in detail.

2.5.1 Converting User Information from UADS

The TSO/E RACONVRT command allows you to convert user IDs from UADS to the RACF database. You can convert some or all of your users by specifying the appropriate operands on the command. See *TSO/E Version 2 System Programming Command Reference* for more information on the RACONVRT command.

RACONVRT generates the RACF commands needed to update or create RACF user profiles and places these commands into members of a partitioned data set. This data set is called “prefix.IKJ.RACONVRT.CLIST,” where “prefix” is the user ID of the person executing the command. RACONVRT creates, but does not execute, the RACF commands. This is to allow you to edit and change the clists to customize the conversion process before you actually issue the RACF commands.

Figure 6 lists the members created by the RACONVRT command, with a short description of each member.

Member	Description
ADDUSER	Defines users to RACF, defines a TSO segment, and creates a data set generic profile.
ALTUSER	Defines a TSO segment within existing RACF user profiles.
DEFAUTH	Defines TSO/E authorities (OPER, ACCT, JCL, MOUNT, and RECOVER) as RACF resources and gives users authority to access these resources
DEFPROC	Defines logon procedures as RACF resources and gives users authority to access these resources.
DEFACCT	Defines account numbers as RACF resources and gives users authority to access these resources.
DEFPERF	Defines performance groups as RACF resources and gives users authority to access these resources.
RUN	Invokes the other members in the proper order to complete the conversion to the RACF data base.

Figure 6. Members Created by the RACONVRT Command

When you are ready to issue the RACF commands, do the following:

1. Issue the SETROPTS command to activate the TSO-related RACF resource classes:

   ```
   SETROPTS CLASSACT(TSOPROC ACCTNUM PERFGRP TSOAUTH)
   ```

2. Execute the RUN member of the clist data set to issue the RACF commands generated by RACONVRT.

3. Issue the SETROPTS command with the RACLIST operand for the newly activated TSO resource classes to bring the resource profiles into storage:

   ```
   SETROPTS RACLIST(TSOPROC ACCTNUM PERFGRP TSOAUTH)
   ```

   When you change any user information in any of these resource classes, you must issue the SETROPTS RACLIST command with the REFRESH option to make the changes active.

2.5.2 Testing the Conversion

To test the conversion from the UADS to the RACF database, log off TSO/E and log back on TSO/E.

2.5.3 Deleting User Information from UADS

After you have tested the conversion and are sure that it was successful, you can delete information from SYS1.UADS for user IDs converted to the RACF database. To delete the user information from the UADS, use the DELETE subcommand of ACCOUNT.

Do not delete any information from the UADS until each user logs on TSO/E for the first time after the conversion. User information stored in the UADS must be available to obtain a user′s current User Profile table (UPT), because the conversion does not migrate the UPT to the RACF database.

Note: Moving the TSO/E user information to the RACF database eliminates the need to maintain an entry in SYS1.UADS for each TSO/E user. However, you MUST maintain entries in SYS1.UADS for certain users (such as IBMUSER and system programmers). For example, if you have to deactivate RACF to perform maintenance on the RACF database, users authorized to perform this maintenance must be able to log on the system. With RACF inactive, TSO/E checks entries in SYS1.UADS to authorize access to system.

2.5.4 Synchronizing RACF Database with Broadcast Data Set

To synchronize the RACF database with the broadcast data set (SYS1.BRODCAST), user either the SYNC command or the SYNC subcommand of the ACCOUNT command.

If you have not converted all user information to the RACF database (as recommended), specify the BOTH operand when you issue the SYNC command. The BOTH operand causes the broadcast data set to be synchronized with both the UADS and the RACF database; that is, for every user ID defined in either the UADS or the RACF database, a corresponding entry is made in the broadcast data set.

Figure 7 shows an example of a job that synchronizes the broadcast data set with the UADS and the RACF database.

```
//jobname   JOB  jobcard parameters
//STEP1     EXEC PGM=IKJEFT01
//SYSTSPRT  DD SYSOUT=A
//SYSRACF   DD DSN=dsname of RACF database,DISP=SHR
//SYSUADS   DD DSN=SYS1.UADS,DISP=SHR
//SYSLBC    DD DSN=SYS1.BRODCAST,DISP=SHR
//SYSTSIN   DD *
  SYNC BOTH
/*
```

Figure 7. Synchronizing Broadcast, UADS, and RACF

If you have converted all user information from the RACF database, specify the RACF operand when you issue the SYNC command. The RACF operand causes the broadcast data set to be synchronized with the RACF database; that is, for every user ID defined to TSO/E in the RACF database, a corresponding entry is made in the broadcast data set.

Figure 8 shows an example of a job that synchronizes the broadcast data set with the RACF database.

```
//jobname   JOB  jobcard parameters
//STEP1     EXEC PGM=IKJEFT01
//SYSTSPRT  DD SYSOUT=A
//SYSRACF   DD DSN=dsname of RACF database,DISP=SHR
//SYSLBC    DD DSN=SYS1.BRODCAST,DISP=SHR
//SYSTSIN   DD *
  SYNC RACF
/*
```

Figure 8. Synchronizing Broadcast and RACF

For more information about the conversion from SYS1.UADS to RACF database see *TSO Extensions Version 2 - Customization*.

2.6 TSO Segment of RACF User Profiles

When you define a new TSO/E user or change TSO/E attributes in the RACF database for an existing user, you can specify the following fields in the TSO segment of a user′s profile:

- ACCTNUM - User′s default account number
- JOBCLASS - User′s default job class
- MSGCLASS - User′s default message class
- HOLDCLASS - User′s default hold class
- SYSOUTCLASS - User′s default SYSOUT class
- DEST - User′s SYSOUT data set destination id.
- PROC - User′s default logon procedure
- MAXSIZE - User′s maximum region size
- SECLABEL - User′s current security label
- SIZE - User′s default region size
- UNIT - Default device(s) used for allocations
- USERDATA - Optional user data

If the user information is stored in the RACF database, you can use a combination of the FIELD class and the command processors so that the RACF administrator can decide which fields users can define and update in their TSO segment. The profile name must be USER.TSO.fieldname. With field-level access checking you can:

- Allow all users to modify a particular field (or segment) of their own user profiles. To do this, specify ID(&RACUID) on the PERMIT COMMAND.
- Allow a user or group to modify a particular field (or segment) in all profiles of a particular type. For example, you can define a profile to control access to the ACCTNUM field of the TSO segment of user profiles. If you give a user Update authority to this profile, the user can modify the ACCTNUM field in all user profiles.

If your installation decides to keep TSO user information in the user profile of the RACF database instead of UADS, you should use RACF to protect TSO/E resources such as logon procedures and account numbers. Refer to 4.9.3, “RACF and TSO/E” on page 52 for information about using RACF to protect TSO/E resources.

Chapter 3. Data Sets

RACF can protect VSAM data sets, cataloged and uncataloged non-VSAM DASD data sets, data sets managed by Storage Management Subsystem (DFSMS*), tape data sets with standard labels, data sets that have the same name but reside on different volumes, and generation data group (GDG) data sets.

To protect tape and DASD data sets, create profiles in the DATASET class by issuing the ADDSD command, or by specifying the PROTECT=YES operand in the JCL or the PROTECT operand on the TSO ALLOCATE command.

You can also use ISPF panels to create profiles in the class DATASET. See Section 9.1, "Data Set Profiles" on page 92 for examples of using these panels.

A data set can be password protected. If the data set has both RACF and password protection, the password protection is bypassed. RACF provides more security than password protection: With RACF protection, only authorized users can access the data set; with password protection, any users who knows the password can access the data set.

When a user attempts to access a data set, RACF verifies both the user and the data set profiles to determine whether to grant or to deny the access.

3.1 Data Set Profile

When you define a data set to be protected, RACF creates a data set profile that contains, among other information:

- The profile name
- The profile owner
- The access list
- The universal access authority (UACC)
- Auditing information
- Erase-on-scratch option

3.1.1 Profile Name

Data set profiles can be either discrete or generic. To create a data set profile, the high-level qualifier has to be a user or a group defined to RACF. Use the Naming Convention table for the exceptions. The Naming Convention table enables an installation to have naming conventions different from RACF standard. For more information regarding this subject refer to *MVS Storage Management Library: Data Sets*.

A discrete profile protects a single data set in a specific volume. In this case, the name of the profile must exactly match the name of the data set it protects. For example, the discrete data set profile SIMON.PAYROLL.DATA protects only the data set SIMON.PAYROLL.DATA in the volume specified on the profile. When a data set that is protected by a discrete profile is deleted, the discrete profile is automatically deleted. Use a discrete profile to protect a data set with unique security or logging requirements.

Whenever possible, use generic profiles for data set profiles. One generic profile protects many data sets on different volumes and devices. With generic profiles, an installation reduces the number of RACF data set profiles, and consequently, the size of the RACF data base.

A generic profile protects several data sets that have a similar naming structure and that have the same security and logging requirements. These data sets must have some identical characters in their names. The name of a generic profile can contain generic characters (*, **, or %) that match any other characters. The high-level qualifier of the profile name must not contain a generic character.

For example, the generic profile SIMON.* could protect the data sets SIMON.DATA, SIMON.DATA.TEST, and SIMON.CNTRL.DATA. See Section 3.2.2, "Creating a Generic Profile" on page 26 for more details.

You can also create a fully qualified generic profile, whose name must match the name of the data set being protected. In spite of this equality, a fully qualified generic profile differs from a discrete profile. A fully qualified generic profile is not deleted when the data set is deleted. Also, with a fully qualified generic profile, you can have multiple data sets in different volumes with the same name, all protected by the same profile.

If a data set matches more than one generic profile, the most specific profile sets the level of protection for the data set. If a data set is protected by both a generic and a discrete profile, the discrete profile sets the level of protection for the data set.

3.1.2 Profile Owner

When you define a data set profile to RACF, a RACF-defined user or group must be defined as the owner of the profile. If you do not explicitly define the owner, RACF names you (the issuer of the command) as the owner of the profile.

The owner has full control over the profile, including the access list. If the owner is a group, users with Group-special authority in that group have also full control over the profile. However, the owner cannot automatically access the data sets protected by the profile. To access these data sets, the owner must be authorized in the profile′s access list, unless the high-level qualifier of the profile name is the owner′s user ID.

3.1.3 Access List

Each data set profile contains a list of specific users and groups who may use the data set(s) protected by the profile, and how they may use the data set(s).

Use the PERMIT command to create entries in the access list. The access authority to use the data set(s) can be one of the following:

NONE Does not allow users to access the data set.

READ Allows users to read-only access the data set. to access the data set for reading only. When you permit users to read a data set, they can copy or print it.

UPDATE Allows users to read from, copy from, or write to the data set. However, they are not authorized to delete, rename, move, or scratch the data set.

CONTROL For non-VSAM data sets, this is equivalent of UPDATE.

For VSAM data sets, CONTROL allows users to access the control-interval, and to retrieve, update, insert, or delete records in the data set.

ALTER Allows users to read, update, delete, rename, move, or scratch the data set.

When specified in a generic profile, ALTER gives users no authority over the profile itself. When specified in a discrete profile, ALTER allows users to read, alter, and delete the profile, but they may not change its owner.

EXECUTE Allows users to load and execute, but not read or copy, load modules (programs).

3.1.4 Universal Access Authority

The universal access authority (UACC) is the default access authority that RACF gives to all users and groups not specified in the profile′s access list.

The levels of access authority that can be defined as the UACC are the same as those specified in the access list. Use the UACC to implement the security requirements for the majority of the users, and specify the exceptions in the access list.

When you specify the UACC in a profile, that UACC applies to all users on the system, both RACF and non-RACF defined. However, RACF 1.9 allows you to restrict data sets access only to users who are defined to RACF. To allow only RACF defined users to access a data set, specify ID(*) on the PERMIT command, and a UACC of NONE for the profile.

3.1.5 Auditing Information

RACF allows you to audit the use of each data set. You can specify the type of access that you want to be recorded by RACF. For example, you can set up a resource profile for your data set to audit every attempt to access the data set, or to audit only the unsuccessful attempts to update the data set.

3.1.6 Erase-on-scratch Option

This option can be used to specify that the data set protected by this profile is to be physically erased when the data set is deleted. When the data set is deleted data management overwrites the contents of all allocated extents with binary zeroes.

To use erase-on-scratch, specify the ERASE operand when defining a data set profile and activate the erase-on-scratch option with the SETROPTS command. See Section 5.2, “Recommended SETROPTS Options” on page 54 for information about the ERASE option in the SETROPTS command.

3.2 Creating Profiles

This section describe how to create simple generic and discrete data set profiles. It shows some examples of commands used to define and maintain a data set profile. For additional information, refer to *RACF General User’s Guide* and *RACF Command Language Reference*.

In addition to RACF commands, you can also use ISPF panels to maintain data set profiles. Refer to Section 9.1, “Data Set Profiles” on page 92 to see some of these panels.

3.2.1 Creating a Discrete Profile

To create a discrete profile, use the ADDSD command. Remember that discrete profiles protect only one data set. If the data set is not cataloged, you must specify the unit type and the volume serial number where the data set resides.

For example, to create a discrete profile for the uncataloged data set NOELY.ACC.DATA, enter the following command:

```
ADDSD ’NOELY.ACC.DATA’ UACC(NONE)
      UNIT(3380) VOLUME(MVS084)
```

You can also use RACF to notify a specific user when an access to a data set is denied by RACF, by using the NOTIFY operand.

For example, to create a discrete profile for the cataloged data set SIMONE.MUG.CHARTS, and to notify the user MILLER when RACF denies access to the data set to any user, enter the following command:

```
ADDSD ’SIMONE.MUG.CHARTS’ UACC(NONE) NOTIFY(MILLER)
```

If you want to allow all the users connected to the group GMPG to read the data set SIMONE.MUG.CHARTS, you have to add this group to the access list in the data set profile. Do this through the command PERMIT, as follows:

```
PERMIT ’SIMONE.MUG.CHARTS’ ID(GMPG) ACCESS(READ)
```

3.2.2 Creating a Generic Profile

By creating generic profiles you can protect more than one data set with a unique profile. You create a generic profile for a data set in the same manner as a discrete profile, except that you include one or more generic characters (*, **, or %), or you include the GENERIC keyword on the ADDSD command.

How you specify the generic characters depends on whether your installation has enhanced generic naming (EGN) active or not.

The generic characters you can use are:

***** With EGN (EGN in effect)

- As a character at the end of a profile name (AB.CD*), matches zero or more characters until the end of the qualifier.
- As a qualifier at the end of a profile name (AB.CD.*), matches one qualifier.
- As a qualifier in the middle of a profile name (AB.*.CD), matches one qualifier.

- As a character at the end of a qualifier in the middle of a profile name (AB.CD*.EF), matches zero or more characters until the end of the qualifier.

Without EGN (NOEGN in effect)

- As a character at the end of a profile name, matches zero or more characters, zero or more qualifiers, or both.
- As a qualifier at the end of a profile name, matches one or more qualifiers.
- As a qualifier in the middle of a profile name, matches any one qualifier.
- As a character at the end of a qualifier in the middle of a profile name, matches zero or more characters until the end of the qualifier.

****** A double asterisk matches zero or more qualifiers. It is allowed only when enhanced generic naming (EGN) is active.

% A percent sign matches one and only one character.

You cannot specify any generic character in the high-level qualifier of a profile in the DATASET class. Also, a single asterisk can appear only as the last character in the qualifier. Therefore, the data set profile names *.BACKUP.DATA and *BACKUP.DATA would be invalid.

Figure 9 shows some generic profile names and examples of resource names that could be covered by profiles with and without EGN active.

Activate EGN as soon as possible and do not deactivate it. If you protect resources with generic profiles while enhanced generic naming is active and then deactivate this option, your resources may no longer be protected, especially when you define a profile that ends with an asterisk. Enhanced generic naming changes the meaning of the * character.

To activate enhanced generic naming, use the SETROPTS command:

```
SETROPTS EGN
```

For example, you can create a generic profile to protect all data sets of the group SALES that are not protected by a more specific profile. If your system does have enhanced generic naming active, you can enter the command:

```
ADDSD 'SALES.**' UACC(NONE)
```

RACF provides the ability to allow access to specified data sets only while executing a certain program. By specifying the operand WHEN(PROGRAM(program-name) on the PERMIT command, you create a conditional access list for the profile protecting the data sets.

For example, to allow user ANGELO read access to data sets protected by the profile SALES.STATUS.* only when executing program STMONTH, enter the following command:

```
PERMIT 'SALES.STATUS.*' ID(ANGELO)
       ACCESS(READ) WHEN(PROGRAM(STMONTH))
```

	Resources Protected	
Profile Name	EGN on	EGN off
AB.CD	AB.CD	AB.CD
AB.CD.%	AB.CD.E AB.CD.F	AB.CD.E AB.CD.F
AB.%D	AB.XD AB.YD	AB.XD AB.YD
AB.CD*	AB.CD AB.CDEF	AB.CD AB.CDEF AB.CD.EF
AB.CD.*	AB.CD.EF	AB.CD.EF AB.CD.EF.GH
AB.*.CD	AB.YZ.CD	AB.YZ.CD
AB.CD*.EF	AB.CD.EF AB.CDYZ.EF	AB.CD.EF AB.CDYZ.EF
AB.**	AB AB.CD AB.CD.EF	Profile invalid
AB.**.CD	AB.CD AB.YZ.CD AB.WX.YZ.CD	Profile invalid

Figure 9. Generic Profiles with and without EGN Active

You can activate generic profile checking for data sets by using the GENERIC operand of the SETROPTS command, for the DATASET class:

```
SETROPTS GENERIC(DATASET)
```

Whenever you create a new generic profile, you should use the REFRESH operand in order to make the new profile effective for already active TSO users, jobs, or started tasks:

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

3.2.3 Creating a Profile For a Tape Data Set

You protect a tape data set by creating a generic or discrete data set profile.

Use the ADDSD command with the TAPE operand to define a profile to protect data sets on tape. If the tape data set to be protected by a data set profile is not cataloged, you must specify the UNIT, VOLUME, and FILESEQ operand when creating the profile.

For example, to create a discrete data set profile to protect the data set SIMON.NEW.DATA residing on tape volume 123456 with a file sequence of 1, enter the following command:

```
ADDSD ′SIMON.NEW.DATA′  TAPE  UNIT(TAPE)
      VOLUME(123456) FILESEQ(1)  UACC(NONE)
```

You can also protect a tape volume. Refer to Section 4.3, “Protecting Data on Tape” on page 38 for more information about tape data sets and tape volumes protection.

3.2.4 Determining Profile Protection

If you want to find out how protected a data set is, you can use the LISTDSD command. This command displays a listing for the profile that is protecting the data set. For example, if you are the owner of the data set MARY.PROJECT.TEXT, you may know that the data set is protected by a generic profile. By issuing the following command, a listing for the profile protecting the data set will be displayed on your screen. The ALL operand specifies you want all information about the profile, including the access list:

```
LISTDSD DATASET(′MARY.PROJECT.TEXT′)  ALL GENERIC
```

To determine if there is a discrete profile protecting the data set, do not use the GENERIC operand. In this way, a listing for the discrete profile that is protecting the data set will be shown:

```
LISTDSD DATASET(′MARY.PROJECT.TEXT′)
```

In either case, if the data set is protected by a fully qualified generic profile, this profile will be displayed.

You can also use the LISTDSD command with the DSNS operand to list all cataloged data sets that are protected by a profile. For example, for the generic profile SALES.*, enter:

```
LISTDSD   DATASET(′SALES.*′)  DSNS
```

The RACF SEARCH command is used to obtain lists of RACF profiles, users, and groups. You can request, for example, a list of profile names that contain a specific character string, or a list of profiles for resources that have not been referenced for more than a specific number of days.

You must have a sufficient level of authority for each profile selected as the result of your request. To search information on profiles within the scope of a group, you must have either Group-special or Group-auditor authority. If you have System-special or System-auditor authority you may use the SEARCH command to seek information in all profiles.

You do not need Special or Auditor authority to use the SEARCH command for profiles you own. For example, to find out what data set profiles you have, enter the command:

```
SEARCH
```

If user ADMIN wants to obtain a list off all data set profiles (both discrete and generic) that have the word BACKUP as the second-level qualifier, and ADMIN has Special authority, he or she can use the following command:

```
SEARCH   FILTER(**.BACKUP.**)
```

The SEARCH command can be used to list all generic profiles that have the potential to protect the data set. The profiles will be listed in the order that RACF uses them. Because all data set profiles begin with a user ID or group name, you can use the FILTER operand to show only those profiles that could protect a data set:

```
      SEARCH   FILTER(userid.**)
 or
      SEARCH   FILTER(groupid.**)
```

You can specify the CLIST operand in the SEARCH command. Specifying this operand, causes RACF to create a CLIST with some commands depending on the other operands in the SEARCH command. This facility makes the RACF data base administration easier.

For example, user ADMIN, who has Special authority, wants to revoke all user IDs that have not accessed the system in the last 90 days. Issuing the following command causes RACF to create a CLIST that can be executed to revoke the users:

```
      SEARCH   CLASS(USER)   AGE(90)
               CLIST(′ALTUSER ′ ′ REVOKE′)
```

Note that the INITSTATS option must be in effect for RACF to record the statistics available during the RACINIT processing.

If all users, except users OLD01, OLD02, and OLD03, have accessed the system in the last 90 days, the resulting CLIST will be:

```
      ALTUSER OLD01 REVOKE
      ALTUSER OLD02 REVOKE
      ALTUSER OLD03 REVOKE
```

The data set userid.EXEC.RACF.CLIST contains the resulting CLIST generated by the SEARCH command. Use the TSO EXEC command to execute the CLIST:

```
      EXEC   EXEC.RACF.CLIST
```

The CLIST operand in the SEARCH command can be used, for example, to create a generic data set profile for each user defined to RACF, with the high level qualifier equal to the user ID. You can enter the following command:

```
      SEARCH   NOMASK  CLASS(USER)
               CLIST(′ADDSD ′ ′.* GEN UACC(NONE)′)
```

If your RACF users are EINSTEIN, THELMA, LOUISE, BART, HOMMER, and MARGIE, the resulting CLIST will be:

```
      ADDSD BART.* GEN UACC(NONE)
      ADDSD EINSTEIN.* GEN UACC(NONE)
      ADDSD HOMMER.* GEN UACC(NONE)
      ADDSD LOUISE.* GEN UACC(NONE)
      ADDSD MARGIE.* GEN UACC(NONE)
      ADDSD THELMA.* GEN UACC(NONE)
```

Be sure to issue the TSO/E command PROFILE NOPREFIX before executing this CLIST.

Optionally, the command SEARCH can be issued through ISPF panels. See Section 9.1.4, “Searching for Data Set Profiles” on page 99 for an example of the use of ISPF panels to search for data set profiles.

3.3 Protecting System Data Sets

MVS system data sets can be divided into two categories, depending on whether the RACF protection is bypassed or not when the system accesses those data sets.

RACF protection for some data sets is bypassed when the system accesses them to perform its normal system functions. However, when a user attempts to access these data sets for normal data set operation, RACF protection is enforced.

SYS1.LINKLIB, SYS1.CMDLIB, SYS1.LPALIB, SYS1.PROCLIB, SYS1.MANx, and other are included in this category. You can specify a UACC of NONE for those data sets and permit update authority to the system programmers responsible for maintaining the data sets. A UACC of NONE does not prevent users from executing any program in the SYS1.LINKLIB, for example, but it does prevent them from opening the data set SYS1.LINKLIB, or specifying it as part of JOBLIB or STEPLIB.

Other data sets, such as SYS1.BRODCAST, SYS1.HELP, and SYS1.MACLIB, are RACF protected when the system accesses them for for its normal system operation on behalf of a specific user. In this case, the user who attempts to access the data set through a system function must have sufficient authority to access the data set.

Data set SYS1.BRODCAST, for example, is opened for update by the TSO SEND command. Therefore, you should give all users update authority to this data set. This is a good candidate to be placed in the Global Access Checking table, with access authority of update. See Section 6.1, “Global Access Checking Table” on page 59 for details about this table.

3.4 Protecting LLA PARMLIB data sets and LLA-managed data sets

Library lookaside (LLA) is an MVS service that maintains a copy of the PDS directory entries for a specified group of production libraries in the LLA address space′s virtual storage. This group of libraries is defined by the installation and includes the set of LNKLST data sets and those specified in the parmlib member CSVLLAxx.

Library lookaside uses virtual lookaside facility (VLF) to keep the most active LLA-managed modules in a data space to avoid program fetch I/O.

You should define profiles in the DATASET class for PARMLIB data sets that contain CSVLLAxx member(s) that specify which libraries LLA is to manage. Permit operators read access to these data sets.

You also should define RACF profiles (either in the FACILITY or the DATASET class) for LLA-managed data sets. These data sets are the libraries that are specified in the CSVLLAxx and LNKSTxx members of a parmlib. Update access is required to the profiles protecting LLA-managed data sets in order to operators to be able to revise the current list of LLA-managed data sets.

3.5 Forcing Cataloging

To encourage cataloging of all data sets, the installation can issue the SETROPTS command with the CATDSNS operand. With this option active, RACF will not allow users without Special authority to access any permanent uncataloged data set, with the following exceptions:

- The data set is protected by a discrete profile.
- The data set is protected by a fully qualified generic profile.
- The user has read access to a profile named ICHUNCAT.dsname in the FACILITY class.

By enforcing cataloging of all data sets, you can use the RACF LISTDSD command with the DSNS option to list all data sets covered by a profile, since these data sets will be found through the catalog. See 3.2.4, "Determining Profile Protection" on page 29 to information about the LISTDSD command.

CATDSNS covers only those data sets that are cataloged in the master catalog or have an alias in the master catalog pointing to a user catalog. When CATDSNS is active, users are not allowed to use JOBCAT and STEPCAT. Create a profile named ICHUCAT in the FACILITY class in order to allow users with read access in this profile to use JOBCAT and STEPCAT.

To activate CATDSNS, enter the following command:

```
SETROPTS  CATDSNS(WARNING)
```

The optional operand WARNING can be used to allow accesses, but send a warning message to the user and the security administrator.

3.6 Data Set Access Checking

This section briefly describes the procedures RACF performs when an access to a data set is requested.

RACF grants or denies the access based on variables, such as the kind of access the requester wants, the requester′s authority, and the UACC of the profile protecting the data set.

Figure 10 shows the basic steps performed by RACF when it is invoked to control the access to a data set.

The first step, Resource in the GAC, refers to the Global Access Checking table, described in Section 6.1, "Global Access Checking Table" on page 59. This book does not cover the RACF Security Classification facility; refer to *RACF Security Administrator′s Guide* for information about this subject.

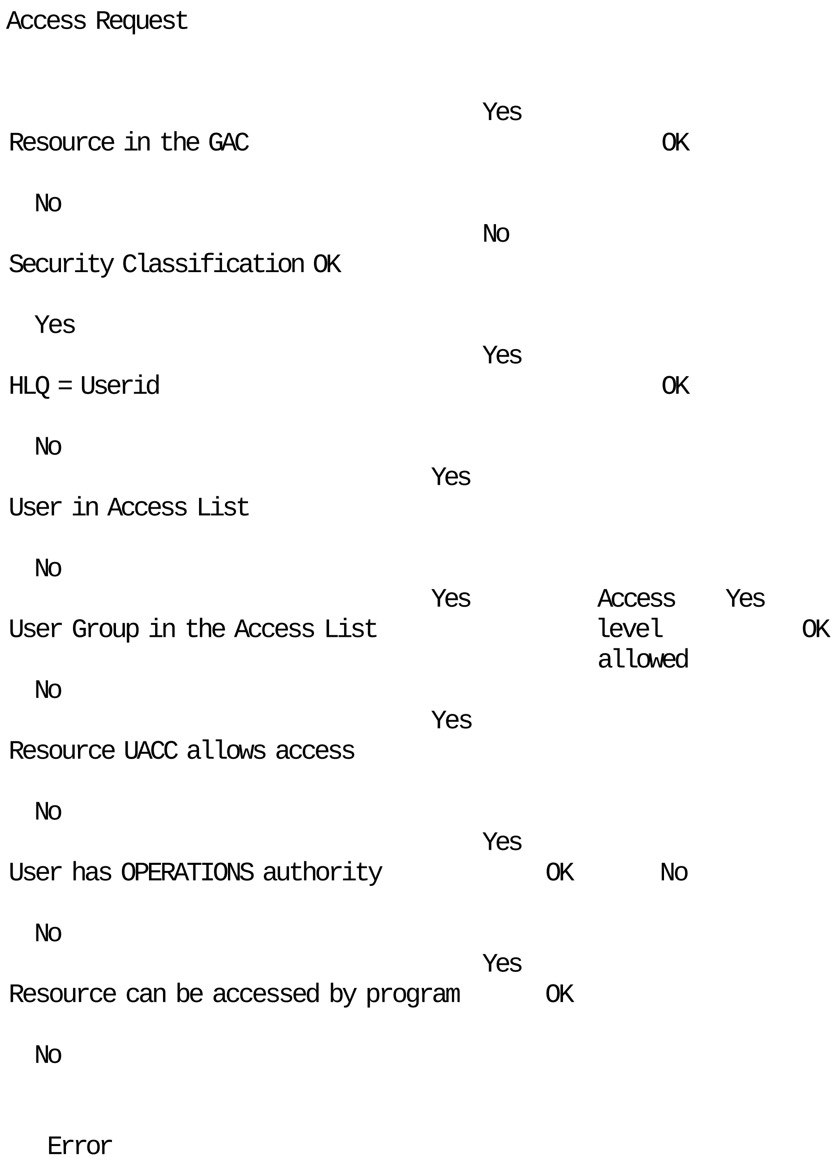

Access Request
Yes
Resource in the GAC
OK
No
No
Security Classification OK
Yes
Yes
HLQ = Userid
OK
No
Yes
User in Access List
No
Yes
Access
level
allowed
Yes
User Group in the Access List
OK
No
Yes
Resource UACC allows access
No
Yes
User has OPERATIONS authority
OK
No
No
Yes
Resource can be accessed by program
OK
No
Error

Figure 10. Data Set Access Checking

Chapter 4. General Resources

Besides protecting the data in DASD and tape data sets, RACF can also protect general resources. General resources are all the resources that are defined in the class descriptor table and include, among others, DASD volumes, tape volumes, terminals, and load modules (programs).

4.1 General Resource Profile

When a resource that is not a DASD data set or a tape data set is defined and protected, RACF builds a general resource profile that contains information about the resource such as class name, resource name, the owner of the profile, an access list and which attempts, successes or failures, are to be logged.

General resources are grouped in classes. When you create a general resource profile, you must specify which general resource class the profile is in. Not all classes are described in this book.

Like data set profiles, general resource profiles can be generic or discrete. If you specify a generic profile name, the profile can protect more than one resource. The resource names contain patterns or character strings that RACF can use to associate them with each other. Using generic profiles instead of discrete profiles can greatly reduce the effort of profile maintenance. In general, you should create generic profiles for the majority of resources, and use discrete profiles only for exceptions.

Unlike data set profiles, when you define a general resource, the RACF command SETROPTS EGN has no effect, as it applies to only data set profiles. For general resource profiles, enhanced generic naming is always active.

To create general resource profiles, use the RDEFINE command. Universal access authorities for a general resource profile can be specified with the UACC operand:

```
RDEFINE class-name  profile-name
        UACC(access-authority)   other-operands
```

If specific users or groups are to have specific access to the profile, use the PERMIT command to create the access list.

```
PERMIT profile-name  CLASS(class-name)
       ID(user or group) ACCESS(access-authority)
```

For general resources, in the same way as for data sets, RACF 1.9 provides the ability to restrict resource access only to RACF defined users. When you specify the UACC in a resource profile, that UACC applies to all users on the system, both those defined and those not defined to RACF. You have the option of using the ID(*) specification on the PERMIT command to give RACF defined users the access you desire, while restricting undefined users to the access specified in the UACC.

Alternatively, you can use ISPF panels to maintain general resource profiles. For details, see Section 9.2, “General Resource Profiles” on page 99.

4.1.1 General Resource Group Profile

For some kinds of resources, consider using resource group profiles instead of generic profiles. Like generic profiles, resource group profiles enable you to protect several resources with one profile. However, the resources do not have to have similar names. Subsystems that have high performance requirements, such as CICS/VS and IMS/VS, have the profile resident in the subsystem address space. These subsystems can save more storage by using resource group profiles. Resources such as terminals and DASD volumes are eligible for resource group profiles.

To create the resource group profile, use the RDEFINE command as follows:

```
RDEFINE resource-group-class  profile-name  UACC(access-authority)
        ADDMEM(resource-name, ...  )
```

where *resource-name* is the name of the resource to be protected, with or without generic characters. The name of a resource group profile (profile-name) cannot contain generic characters (*, **, and %).

For an example of the use of resource group profile, refer to Section 4.4, "Protecting Terminals" on page 41.

4.1.2 Using RACFVARS to Protect General Resources

Starting with RACF 1.9, you can create profiles in the RACFVARS class whose profile names act like programming variables (indicated by an &). Like resource group classes, you can use the RACFVARS class to create general resource profiles that protect many resources with unlike names.

The use of RACFVARS reduces the number of profiles that are needed. Variables cannot be used in data set profile names.

To create a profile in the RACFVARS class, specify the resource names on the ADDMEM operand:

```
RDEFINE RACFVARS  profile-name  UACC(READ)
        ADDMEM(resource-name, ...  )
```

The profile name must begin with the character &. Note that the RACFVARS class must be in the RACLIST to take effect.

See Section 4.3.2, "Protecting Tape Volumes Using the RACFVARS Class" on page 40 for an example of the use of the RACFVARS class.

If you have resources with similar names, you should still use generic profiles. When available, resource group classes should still be used (GTERMINL, for example).

4.1.3 Activating the Protection

When you are ready to start using the protection defined in the profiles, activate the general resource class using the SETROPTS command with the CLASSACT operand for each class:

```
SETROPTS CLASSACT(class-name)
```

Only the DATASET class is active when RACF is initialized for the first time.

For performance benefits, consider implementing one of the following:

- Allow all users on the system to have access to the resource at some level (such as read) by creating a Global Access Checking table entry. See 6.1, “Global Access Checking Table” on page 59 for more information about this subject.
- Reduce I/O to the RACF data base by requesting that RACF keep all profiles in the class in storage. Use the SETROPTS command with the RACLIST operand to request that function. When activating specific general resource classes, such as RACFVARS, OPERCMDS, and others, you must have these classes in the RACLIST.

If you have the Special authority, you can activate or deactivate generic profile checking for a specified class. You can specify this option with the GENERIC and NOGENERIC operands of the SETROPTS command:

```
SETROPTS GENERIC(class-name)
```

If you create a new generic profile after activating generics for a class, you should re-enter the SETROPTS command with the REFRESH operand. If you do not, the new profile will not be used for TSO users who are currently logged on or for active jobs or started tasks:

```
SETROPTS GENERIC(class-name) REFRESH
```

For additional details on the SETROPTS command, refer to Section 5.1, “SETROPTS Command” on page 53.

4.2 Protecting DASD Volumes

Some maintenance operations have to be performed on a DASD volume, such as initialize, dump, restore, and scratch. The user must have the necessary authority in the DATASET class to perform those operations on each of the data sets on the volume.

By defining profiles in the DASDVOL class, you can define DASD volumes to RACF and authorize users to access the DASD without having access to the DATASET profiles protecting the data sets, depending on the product that the user is using. For example:

- Using DADSM to scratch data sets on the volume requires alter access authority in the DASDVOL class.
- Using Device Support Facilities (ICKDSF), alter authority allows the user to rename DASD volumes.
- Using Data Facility Data Set Services (DFDSS), the access authority required depends on the specific action the user is requesting.
- Other products can also check for authorization in the DASDVOL class.

  Note: DASDVOL authority does not allow users to work with SMS-managed volumes.

A user with Operations or Group-operations authority can also perform those maintenance functions. Using DASDVOL is more efficient for functions such as volume dumping, because only one authorization check (RACHECK) for the volume has to be issued, instead of individual requests for each data set on the volume. Also, DASDVOL allows you to authorize operations personnel to access only those volumes they must maintain.

To activate the DASDVOL class, use the SETROPTS command as follows:

```
SETROPTS CLASSACT(DASDVOL)
```

4.3 Protecting Data on Tape

RACF allows you to establish access requirements for both tape data sets and the tape volume. To protect data on tape, you can do either or both of the following:

- Control access to the tape volume(s) by activating the TAPEVOL class.
- Control access to individual tape data set(s) on the tape volume(s) by activating TAPEDSN class.

There four combination of tape volume (TAPEVOL) and tape data set (TAPEDSN) protection are briefly discussed in this section:

- Tape Data Set and Tape Volume Protection

  Using the ADDSD command for a tape data set results in two discrete profiles: a tape DATASET profile and an automatic TAPEVOL profile containing a TVTOC (tape volume table of contents).

  In this case, RACF uses the TVTOC to provide functions such as:

  – Checking the security retention period (the number of days that a data set is protected against deletion or overwriting).

  – Protection for multiple data sets on a volume.

  – Integrity for full 44-character data set names.

- Tape Data Set Protection Only

  Using the ADDS command with the TAPE operand creates a profile only in the DATASET class. There is no TAPEVOL profile, which means that tape volumes have no RACF protection. Thus, functions such as protection for multiple data sets on a volume and security retention period checking are not performed by RACF.

  In this case, installations depend on their own tape library management system to proceed functions like those.

- Tape Volume Protection Only

  This protection is defined to RACF in one of two ways:

  – By an authorized user issuing the RDEFINE command without the TVTOC operand.

  – By the RACDEF macro when the TAPEVOL class is active and the PROTECT operand is specified on a JCL DD statement, or during EOV processing.

  Since RACF permits a user who has access to a tape volume to access all data sets on that volume, you should place only data sets that requires similar RACF authorization on the same volume.

- No Tape Volume or Tape Data Set Protection

  If you keep both TAPEVOL and TAPEDSN classes not active, data management does not call RACF to check accesses to data on tape. Therefore, you have no RACF protection for data on tape.

Information provided in Chapter 3, “Data Sets” on page 23 applies to both DASD and tape data sets. Following sections describe how to protect tape volumes.

4.3.1 Protecting Tape Volumes

You can provide RACF protection for tape volumes by creating TAPEVOL profiles. Any data set on that tape volume is protected under the profile you created for the volume.

When you define a tape volume to be RACF protected, you can use the RDEFINE command with or without the TVTOC operand. Whatever method you choose, do not forget to activate the TAPEVOL class by issuing the SETROPTS command with the CLASSACT(TAPEVOL) operand.

4.3.1.1 Tape Volumes with a TVTOC

If you specify the TVTOC operand while defining a profile in the TAPEVOL class, RACF creates and maintains a tape volume table of contents (TVTOC). The TVTOC is part of the tape volume profile in the RACF database. The TVTOC contains information such as:

- The number of data sets on the volume.
- The full 44-character name used when creating the data set (from the DSN operand of the JCL DD statement).
- Volume serial number(s) of the volume(s) on which the data set resides (from the VOL operand of the DD statement).
- The file sequence number (from the LABEL operand of the DD statement).
- The RACF internal data set name.

Tape volumes defined with the TVTOC operand can be used by any RACF defined user for writing. When the first user writes a data set on a tape volume with TVTOC, RACF creates an access list for that volume and places that user in the access list with alter authority. Then, only that first user and any users added to the access list with update authority can write additional data sets to the volume.

When a user requests access to a data set on a tape volume with TVTOC, RACF first checks the user′s authority to the volume on which the data set resides. If the user has sufficient authority to the volume, RACF grants access to the data set. If not, RACF searches for DATASET profiles in order to grant or deny the access.

For example, to define a tape volume labeled PR0009 with a TVTOC and assign it a UACC of NONE, enter the following command:

```
RDEFINE TAPEVOL  PR0009  TVTOC  UACC(NONE)
```

4.3.1.2 Tape Volumes without a TVTOC

You can also define profiles in the TAPEVOL class without a TVTOC to control access to data sets on the volume. In this case, only the access list in the volume′s profile is used by RACF when checking whether a user can access a tape data set.

Users with at least read authority to the tape volume can read any data set on the tape. Users with at least update authority to the volume can write data on the tape.

For example, to define a tape volume labeled FI0084 and assign it a UACC of NONE, enter the command as follows:

```
RDEFINE TAPEVOL  FI0084  UACC(NONE)
```

A pool of tape volumes is particularly useful when you want to protect multi-volume tape data sets.

For example, by specifying the command below, you will place the tape volumes labeled FI0084 (already defined) and FI0032 in a pool:

```
RALTER TAPEVOL  FI0084  ADDVOL(FI0032)
```

Since a single profile is maintained in the RACF database for a tape volume set, these volumes share the same access list and the same statistics and auditing options.

In this case, every time you make a change to one of the volumes, the change applies to all volumes in the set. For example, if you give a user update authority to volume FI0084, this user also receives update authority to profile FI0032.

Do not use the RDELETE command to delete a volume in a tape volume set. RACF deletes the definition for all the volumes in the set. To remove a tape volume from the pool, use the RALTER command with the DELMEM operand. For example, to remove the tape volume labeled FI0032 from the pool, issue the following command:

```
RALTER TAPEVOL  FI0084  DELVOL(FI0032)
```

4.3.2 Protecting Tape Volumes Using the RACFVARS Class

Starting with RACF 1.9, qualifiers of general resource profile names can be variables by using a profile in the RACFVARS class. This new facility allows one general resource profile to protect many resources with unlike names.

For example, if your tape volume naming convention does not allow the creation of a generic profile, you can create a profile in the RACFVARS class to assign several values to TAPEVOL profiles.

First, define a RACFVARS profile. The following profile (&TVAR) covers tape volumes TP0001 and S10184:

```
RDEFINE   RACFVARS   &TVAR   UACC(NONE)
          ADDMEM(TP0001 S10184)
```

Then, define the profile in class TAPEVOL and permit some users (the SYSTEMS group, for example) to access those volumes:

```
RDEFINE   TAPEVOL   &TVAR   UACC(NONE)

PERMIT    &TVAR   CLASS(TAPEVOL)
          ID(SYSTEMS) ACCESS(UPDATE)
```

If another tape volume (SS0001) becomes one of the tapes to be protected by &TVAR, issue the following command:

```
RALTER    RACFVARS   &TVAR   ADDMEM(SS0001 SM6452)
```

If a tape no longer has to be protected by &TVAR, it can be removed by using the DELMEM operand in the RALTER command; for example:

```
    RALTER   RACFVARS   &TVAR   DELMEM(SM6452)
```

Figure 11 shows how the single profile &TVAR accomplishes what would have taken three profiles if a RACFVARS profile had not been used.

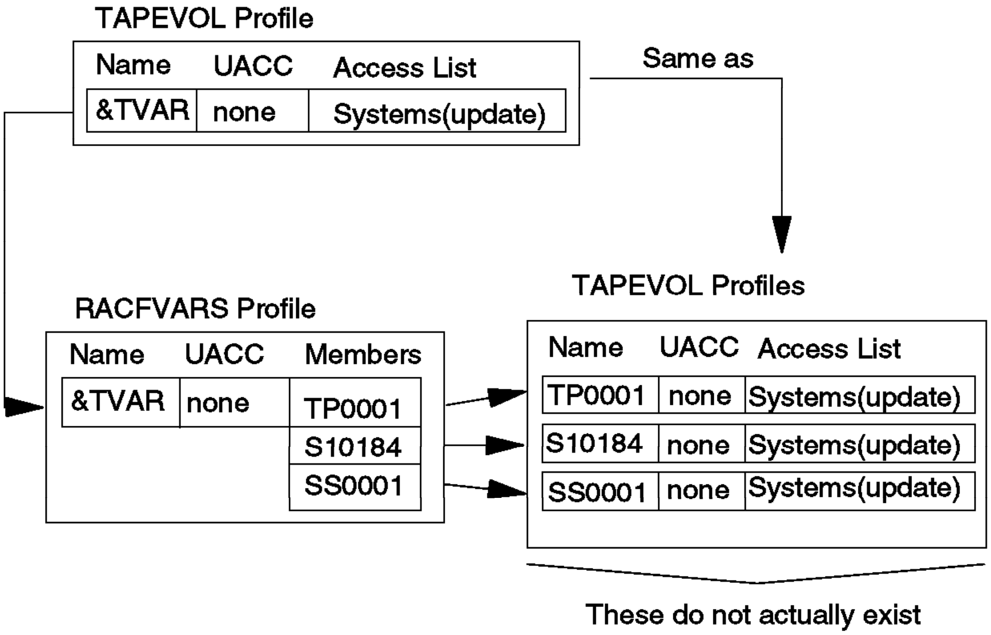


Figure 11. Using RACFVARS to Protect Tape Volumes

Activate the TAPEVOL and RACFVARS classes and enable RACLIST processing for the RACFVARS by using the SETROPTS command, as follows:

```
    SETROPTS  CLASSACT(TAPEVOL RACFVARS)

    SETROPTS  RACLIST(RACFVARS)
```

Use the REFRESH operand in the SETROPTS RACLIST command whenever you change a profile in the RACFVARS class.

4.4 Protecting Terminals

RACF provides several methods of controlling the use of terminals connected to an MVS system. These methods include:

- Creating profiles in the TERMINAL or GTERMINL classes
- Preventing the use of undefined terminals
- Restricting when a terminal can be used
- Restricting specific groups of users to specific terminals

4.4.1 Creating Profiles in the TERMINAL or GTERMINL Classes

If only specific users are to use the protected terminal(s), specify a UACC of NONE. Then, use the PERMIT command to grant access authority to the terminal(s). Users must have at least read access authority to the profile in order to use the terminal(s).

On systems using VTAM or TCAM, the terminal′s node name is the RACF resource name. On systems using BTAM, the resource name for the terminal consists of the relative line and terminal number.

Note: After you define a terminal and protect it with a UACC of NONE, no one can use the terminal until you grant users or groups read access authority to the resource.

If you have terminals that do not allow you to create a generic profile (non-similar names, for example) but need same kind of protection, you can create a resource group profile in the GTERMINL class. For example, to create a profile to protect terminals M01RF267, M03RF168, and M04GG148 you can use the following command, where DEPT35 is a sample profile name:

```
RDEFINE GTERMINL  DEPT35  UACC(NONE)
        ADDMEM(M01RF267 M03RF168 M04GG148)
```

To allow a defined group, FINANCE, to use these terminals, issue the PERMIT command as follows:

```
PERMIT  DEPT35  CLASS(GTERMINL) ID(FINANCE) ACCESS(READ)
```

4.4.2 Preventing the Use of Undefined Terminals

If you have terminals that are not defined to RACF, you can use RACF to control whether they can be used for logging on by issuing the SETROPTS command with the TERMINAL operand:

```
SETROPTS TERMINAL(READ or NONE)
```

When you specify read access authority, users can access all terminals for logging on to the system. To prevent undefined terminals from being used for logging on, specify access authority of NONE. When you specify NONE, only users and groups that are authorized to use a RACF defined terminal in its access list can use it.

Note: Before you specify NONE, be sure that you define some terminals to RACF and give the appropriate users and groups proper authorization to use them. Otherwise, no one will be able to log on to your system.

4.4.3 Restricting When a Terminal Can Be Used

You can use RACF to limit the use of specific terminals to certain days of the week, and certain hours within each day. The WHEN operand on the RDEFINE and RALTER commands for the TERMINAL or GTERMINL classes allows you to control when the system may be accessed from the terminal. For example, to allow logons at terminal M01SS184 only between 8 A.M. and 6 P.M. during the week, but not during the weekend, specify:

```
RDEFINE  TERMINAL M01SS184
         WHEN(DAYS(WEEKDAYS) TIME(0800:1800))
```

4.4.4 Restricting Specific Groups of Users to Specific Terminals

When defining or changing a group profile, you can specify that users in the group can use only those terminals to which they are specifically authorized on the access list in the TERMINAL profile protecting the terminal. Specify option NOTERMUACC for the group on the ADDGROUP or ALTGROUP command:

```
ALTGROUP  PAYROLL NOTERMUACC
```

For example, if you want to restrict group PAYROLL to log on only to terminals in the payroll office, protect them with a profile:

```
RDEFINE  GTERMINL  PAYTERMS
         ADDMEM(M01SS184 M08SM220) UACC(NONE)
```

Then, give the PAYROLL group read access:

```
PERMIT  PAYTERMS  CLASS(GTERMINL)
        ID(PAYROLL) ACCESS(READ)
```

Since the PAYROLL group has NOTERMUACC, users in this group are not allowed to log on to another terminal, even if the terminal has a UACC of READ.

4.4.5 Activating Terminal Protection

When you are ready to activate RACF protection for terminal profiles, activate the TERMINAL class by issuing the SETROPTS command as follows:

```
SETROPTS  CLASSACT(TERMINAL)
```

For performance benefits you can active RACLIST processing for this class. Also, after creating a GTERMINL profile, you must request RACLIST processing for the TERMINAL class (not for the GTERMINL class):

```
SETROPTS  RACLIST(TERMINAL)
```

Be sure to refresh the RACLIST whenever you change profiles in classes already active. RACF will make the changes immediately in effect on the system. For example, if you change profiles in the TERMINAL or GTERMINL classes, enter:

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

4.5 Program Control

Program Control is a RACF facility that can be used to protect load modules (programs), thus providing the installation with the ability to control who can execute what programs. You protect individual programs by creating a profile for the program in the PROGRAM general resource class.

To set up program control, you must protect the library from which the program is loaded. You should protect any library that contains RACF controlled programs with data set profiles that have a UACC of NONE or EXECUTE. This approach prevents unauthorized users from copying programs.

Whit RACF 1.9 you can audit attempted accesses to controlled programs and specify a user ID to be notified whenever RACF uses the program profile to deny access to the program.

4.5.1 Program Access to Data Sets

An installation may wish to make the use of a protected program a condition for access to a specific data set. This features is called Program Access to Data Sets, or PADS. It allows users to access specified data sets at a specified access level only while executing a certain program.

To implement PADS, create a conditional access list for the DATASET profile protecting the data sets, specifying WHEN(PROGRAM(*program-name*)) with the ID and ACCESS operands on the PERMIT command. The user or group specified must be running the specified program to receive the specified access.

The WHEN(PROGRAM) operand on the SETROPTS command activates and deactivates program control. It causes RACF to construct an in-storage table that describes the controlled programs and who can access them.

4.5.2 Protecting Sensitive Programs

When protecting sensitive programs, you should first ensure that the programs are stored in libraries protected by data set profiles. This applies to load modules, JCL, and executive programs. In addition, to protect load modules, you can use RACF′s program control. This allows people to use, but not copy protected programs.

The following sequence of RACF commands illustrates an example of defining a load module as a RACF controlled program. The service utility AMASPZAP, which calls a second module AMAZAP, is an example of a sensitive program whose execution can be protected.

For this example, assume that AMASPZAP and AMAZAP have been placed in SYS1.MIGLIB on the MVSSYS volume, and that their use is to be provided only to the users connected to SYSTEMS group. Protection could be accomplished through the following steps:

1. Define a DATASET profile to protect program library SYS1.MIGLIB. Use a UACC of EXECUTE to limit copying of the programs. Users can execute, but not access the library:

   ```
   ADDSD ′SYS1.MIGLIB′ UACC(EXECUTE) GENERIC
   ```

   When using EXECUTE, you must ensure that all programs in the library are defined to RACF in the PROGRAM class. RACF will not allow the loading of a program from a library with EXECUTE access authorization if the environment into which it is being loaded is not a controlled environment.

2. Define a profile in the PROGRAM class that protects the controlled programs. You can use generic or discrete profiles:

   ```
   RDEFINE PROGRAM AMASPZAP UACC(NONE)
           ADDMEM(′SYS1.MIGLIB′/′MVSSYS′/PADCHK)

   RDEFINE PROGRAM AMAZAP   UACC(NONE)
           ADDMEM(′SYS1.MIGLIB′/′MVSSYS′/PADCHK)
   ```

   Or

   ```
   RDEFINE PROGRAM AMA*  UACC(NONE)
           ADDMEM(′SYS1.MIGLIB′/′MVSSYS′/PADCHK)
   ```

   The PADCHK operand means that RACF checks for program-acessed data sets that are already open before executing the program. If there are any open program-accessed data sets in the address space, RACF ensures, before it allows this program to be loaded, that the conditional access list in the profiles for each data set allows the access. If the access is not allowed, the program is not loaded.

3. Permit the SYSTEMS group access to ZAP:

```
PERMIT AMASPZAP CLASS(PROGRAM) ID(SYSTEMS) ACCESS(EXECUTE)

PERMIT AMAZAP   CLASS(PROGRAM) ID(SYSTEMS) ACCESS(EXECUTE)
```

   Or

```
PERMIT AMA*  CLASS(PROGRAM) ID(SYSTEMS) ACCESS(EXECUTE)
```

4. Activate program control:

```
SETROPTS WHEN(PROGRAM)
```

   Or, if it is already active, refresh the in-storage program control table:

```
SETROPTS WHEN(PROGRAM) REFRESH
```

   This command ensure that the changes will take effect immediately.

4.6 Protecting Temporary Data Sets

Normally, temporary data sets are considered protected from any accesses except by the job or session that created them. However, temporary data sets can be unprotected in the following situations:

- System failure
- initiator failure or initiator terminated by the FORCE command
- automatic restarts during the time between failure and the restart.

Using RACF 1.9 and DFP 3.1.1 or later, you can protect temporary data sets by activating the TEMPDSN class. When this class is active, the only access allowed, besides the owning job during its execution, is by users who have Operations authority. This authority allows users to scratch the data set, but does not allow access to the data set for any other purpose.

To activate the TEMPDSN class, enter the following command:

```
SETROPTS   CLASSACT(TEMPDSN)
```

Note that you cannot create profiles in this class.

4.7 Controlling JES Resources

With the enhancements made to JES2 3.1.3 and JES3 3.1.3 you can control and audit resources managed by JES, using RACF 1.9.

Figure 12 shows the resource classes managed by JES, that can be controlled by RACF.

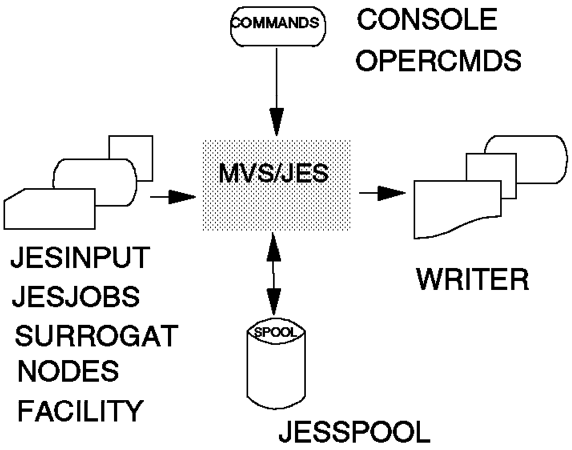


Figure 12. RACF General Resource Classes Related to Job Processing

To activate these classes, issue the SETROPTS command with the CLASSACT operand for each class for which you want to have RACF protection.

RACLIST processing is required for some of these resource classes, such as OPERCMDS and WRITER. The following command is used to activate and RACLIST a resource class:

```
SETROPTS  CLASSACT(class-name)  RACLIST(class-name)
```

Whenever a change is made to one of the profiles in a class that has RACLIST processing, the in-storage profiles must be refreshed for the changes take effect:

```
SETROPTS  RACLIST(class-name)   REFRESH
```

These commands must also be issued on any systems that share the RACF database.

4.7.1 JESINPUT Class

This class permit an installation to limit the input devices that are valid for job submission. Users or groups who have to use a protected input source must have at least read authority to that resource.

It is strongly recommended that you create a profile with UACC of READ for all JES input sources that are not defined. Otherwise, RACF will not allow job submission from those sources. Users will be able to access only JES input sources to which they (or their groups) are explicitly authorized.

For example, to allows all users except BART to enter jobs through the input source RDR2, issue the following commands:

```
RDEFINE  JESINPUT   RDR2  UACC(READ)

PERMIT   RDR2  CLASS(JESINPUT)  ID(BART)  ACCESS(NONE)
```

Before you activate the JESINPUT class using the SETROPTS command, be sure you have created at least one profile for a JES input source for job submission. Otherwise, users will be prevented from submitting batch jobs from any source.

4.7.2 JESJOBS Class

You can use profiles in this class to control who can submit or cancel jobs by job name. It can be used to enforce job name standards. To cancel jobs using the TSO CANCEL command or to submit jobs, users must have at least read access to the profiles protecting the jobs names.

You can specify a UACC of READ for users to submit all jobs and then permit only a select number of users to submit restricted jobs.

For example, to allow all jobs names to be submitted, enter the following command:

```
RDEFINE  JESJOBS   SUBMIT.*.**   UACC(READ)
```

To allow only user ERNIE at node SESAMO to submit a job named MYJOB, enter the following commands:

```
RDEFINE  JESJOBS   SUBMIT.SESAMO.MYJOB.*   UACC(NONE)

PERMIT  SUBMIT.SESAMO.MYJOB.*  CLASS(JESJOBS)
        ID(ERNIE)     ACCESS(READ)
```

To allow a user to cancel a job using the TSO CANCEL command, this user must have alter authority to the profile.

For example, to allow user BERT in SESAMO to cancel jobs starting with SES*, use the following commands:

```
RDEFINE  JESJOBS   CANCEL.SESAMO.*.SES*   UACC(NONE)

PERMIT  CANCEL.SESAMO.*.SES*  CLASS(JESJOBS)
        ID(BERT)   ACCESS(ALTER)
```

Define at least one profile with UACC of READ to allow users to submit jobs before you activate the JESJOBS class.

4.7.3 SURROGAT Class

This class allows an installation to use surrogate users. A surrogate user can submit jobs on behalf of another user (the original user specified on the job card) without having to specify the original users′s password.

Jobs submitted by the surrogate user execute with the authority of the original user (the owner of the job), as though the owner of the job had submitted the job. The output of the job is owned by the original user.

For example, to allow user HOMMER to submit jobs that are owned by user MARGIE, define MARGIE in the SURROGAT class, and then allow HOMMER to submit jobs for MARGIE. Enter the following commands:

```
RDEFINE  SURROGAT    MARGIE.SUBMIT
         UACC(NONE)  OWNER(MARGIE)

PERMIT  MARGIE.SUBMIT   CLASS(SURROGAT)
        ID(HOMMER)   ACCESS(READ)
```

To access the output of the job, the surrogate user (HOMMER, in the example) must be permitted to access the output of the original user (MARGIE) in the JESSPOOL CLASS.

4.7.4 NODES Class

RACF uses this class for security controls when jobs or SYSOUT are received at the system from other nodes. Also, RACF can control whether the jobs entering the system from other nodes need user ID and password verification checking.

This class is also used to translate user IDs, groups, and SECLABELs to locally defined values. It avoids the complexity of defining identical user IDs, group IDs, and SECLABELs in every node in a NJE network.

UACCs, rather than access lists, are used in this class. A UACC of NONE prevents jobs from the specified node from entering your system. Read access allows work from the specified node enter this system, if a user ID and password are supplied. If no applicable profile exists, jobs are treated as though a UACC of READ was specified.

For example, to accept jobs from any user at NEWYORK, with user ID and password verification, create a profile in the NODES class as follows:

```
RDEFINE  NODES    NEWYORK.USERJ.*   UACC(READ)
```

To reject jobs from user MICKEY at ORLANDO, enter:

```
RDEFINE  NODES    ORLANDO.USERJ.MICKEY   UACC(NONE)
```

The use of ** is not valid for the NODES class. You have to use *. If you decide to activate this class, it must be included in RACLIST.

4.7.5 FACILITY Class

Profiles in the FACILITY class can be used to allow resource managers (typically program products or components of MVS) to control the user′s ability to take some action. Access to the profile might be required to perform specific tasks. RACF does not document all the uses of FACILITY-class profiles by other products; you must refer to that product′s documentation. Some examples of the use of these profiles are presented in this section.

If a system supports bypass label processing (BLP) for tape volumes, the FACILITY-class profile ICHBLP allows the installation to control who can use BLP. RACF checks the user′s authority to the ICHBLP profile when the user attempts to access a tape with an IBM standard or ANSI label. You must have the TAPEVOL class active as well for this checking to take place.

If your installation has an IBM processor with the Vector Facility, you can define the profile IEAVECTOR in the FACILITY class in order to control the use of the Vector Facility.

If your installation uses Hiperspace* for performance benefits, RACF can control the use of this resource, through a profile in the FACILITY class. For example, to allow users connected to the group HOTGRP to place their buffers in the Hiperspace named BLSR, issue the following commands:

```
RDEFINE  FACILITY  CSR.BLSRHIPR.BLSR

PERMIT  CSR.BLSRHIPR.BLSR   CLASS(FACILITY)
        ID(HOTGRP)   ACCESS(READ)
```

There are many other functions that can be controlled through the use of the FACILITY class as well, including program dumps, access to uncataloged permanent or system temporary data sets, and many catalog functions, including SMS-related functions.

4.7.6 CONSOLE Class

RACF allows you to specify which MVS consoles operators can use by creating a profile for each console in the CONSOLE class. For protected consoles, RACF ensures that the person attempting to use them has the authority to do so.

For example, to allow only the user OPERAT to use all consoles on the system, create a profile called * in the CONSOLE class with a UACC of NONE and give user OPERAT read access authority, as follows:

```
RDEFINE  CONSOLE  *  UACC(NONE)

PERMIT  *  CLASS(CONSOLE)  ID(OPERAT)  ACCESS(READ)
```

4.7.7 OPERCMDS Class

This class can be used to authorize or to restrict users from issuing some or all MVS and JES commands. If you decide to activate the OPERCMDS class, it must be included in RACLIST.

To ensure that an operator (or group of operators) can issue certain operator commands only when using a particular console, you can create a conditional access list.

For example, to allow user MARC to use all MVS and JES2 commands, but only when using console 05, by entering the following commands:

```
RDEFINE  CONSOLE  *  UACC(NONE)

PERMIT  *  CLASS(CONSOLE)  ID(MARC)  ACCESS(READ)

PERMIT   MVS.*    CLASS(OPERCMDS)  ID(MARC)
         ACCESS(CONTROL)   WHEN(CONSOLE(05))

PERMIT   JES2.*   CLASS(OPERCMDS)  ID(MARC)
         ACCESS(CONTROL)   WHEN(CONSOLE(05))
```

The difference between the OPERCMDS and CONSOLE resource classes is that CONSOLE checks whether a user can log on to a console, while OPERCMDS determines what commands the user is allowed to issue from the console. For a security environment, these two classes should be used together.

4.7.8 JESSPOOL Class

By activating this class you can protect the data sets that reside on the JES spool.

If there is no profile for a data set on the spool, only the user that created the data set can access, modify, or delete it. Authorization to these data sets for other users is provided through profiles in the JESSPOOL class.

For example, to allow user THELMA to read the SYSOUT created by jobs executed by user LOUISE, use the commands below. Note that Read access does not allow THELMA to delete the SYSOUT data sets:

```
RDEFINE  JESSPOOL    NODEID.LOUISE.**
         UACC(NONE)  OWNER(LOUISE)

PERMIT  NODEID.LOUISE.**  CLASS(JESSPOOL)
        ID(THELMA)   ACCESS(READ)
```

4.7.9 WRITER Class

You can use this class to control where output can de printed. You can authorize or restrict the use of writers for local printers, RJE and RJP devices, and network nodes.

For NJE, RACF verifies the security of jobs and SYSOUT transmissions to ensure that the user is authorized to send data to another node in the network.

For example, to prevent SYSOUT from being processed on the local printer PRINTER2, unless the owner of the SYSOUT data set is the SYSTEM group, enter these RACF commands:

```
RDEFINE  WRITER  JESX.LOCAL.PRINTER2   UACC(NONE)

PERMIT  JESX.LOCAL.PRINTER2    CLASS(WRITER)
        ID(SYSTEM)    ACCESS(READ)
```

To allow only user groups connected to group SYSTEM to transmit jobs and SYSOUT to a remote node named SYSBKP, enter the following commands:

```
RDEFINE  WRITER  jesx.NJE.SYSBKP  UACC(NONE)

PERMIT  jesx.NJE.SYSBKP   CLASS(WRITER)
        ID(SYSTEM)   ACCESS(READ)
```

In the example above, **jesx** represents the JES2 subsystem name or the JES3 system name.

4.7.10 Defining Local Nodes

In some resource classes managed by JES, the profile name contains the nodeid as one of its qualifiers. You can define an &RACLNDE profile in the RACFVARS class, where you specify to RACF what nodes are to be considered local. This definition simplifies the maintenance process, specially if the RACF database is shared by two or more systems.

To define to RACF that, for example, nodes LOCALA and LOCALB are to be considered local nodes, use the RDEFINE command to create a profile in the RACFVARS class, as follows:

```
RDEFINE  RACFVARS  &RACLNDE  ADDMEM(LOCALA LOCALB)
```

By creating this profile, you can specify &RACLNDE in place of the nodeid, when the profile has to cover resources for the local node. For example, to allow any user to submit any jobname at a local node, when creating a profile in the JESJOBS class issue the following command:

```
RDEFINE  JESJOBS  SUBMIT.&RACLNDE.**  UACC(READ)
```

The &RACLNDE profile is also required for spool offload and reload with JES 3.1.3 or later, and for other processes. All customers should define the profile and keep it updated.

For complete information on how RACF and JES work together, see *RACF Security Administrator's Guide* and *MVS/ESA and RACF 1.9 Security Implementation Guide*.

4.8 Restricting the Creation of New Profiles

You can restrict the creation of profiles in general resource classes by issuing the RACF option GENERICOWNER.

Prior to RACF 1.9, a user with authority to create profiles in a general resource class (CLAUTH authority) could create any profiles within this class. This user could create profiles more specific than profiles previously created, thereby changing the protection intended by the original profile.

With RACF 1.9, with the GENERICOWNER option in effect, if a generic profile in a general resource class already exists to protect the resource, then a more specific profile to protect the same resource can be created only by a user who has class authorization in that class and is one of the following:

- The owner of the less specific generic profile
- A user with System-special authority
- A user with Group-special authority in the group that owns the less specific profile
- A user with Group-special authority in the group that owns the user that owns the less specific profile.

To activate the GENERICOWNER option, enter the following command:

```
SETROPTS GENERICOWNER
```

Note that this option does not apply to data set profiles. Also, it can not be applied to an individual class; when active, it affects all general resource classes.

4.9 RACF and other Products

This section briefly outlines how RACF protects resources belonging to other programs, such as CICS, IMS/VS, and TSO/E.

4.9.1 RACF and CICS

RACF data set protection can be used to control access to CICS system data sets and program libraries, as well as user files and databases accessed by CICS.

RACF provides a set of classes that can be activated to improve the protection for CICS resources, such as:

- CICS transactions
- CICS File Control table
- CICS Destination Control table
- CICS Program Control table
- CICS Temporary Storage table

CICS/ESA 3.1.1 is the last release to support CICS internal security. An external security program, such as RACF, must be used to provide security functions to control CICS resources.

For more details about using RACF to protect CICS resources, refer to *RACF Security Administrator's Guide*, *CICS-RACF Security Guide*, and the CICS library.

4.9.2 RACF and IMS/VS

Many of the IMS/VS system resources, such as IMS/VS databases and libraries, can be controlled by through the RACF DATASET class.

After establishing RACF protection for IMS/VS data sets, you can use RACF facilities to provide more security for IMS/VS, such as:

- Providing user identification and authentication
- Using operator identification to control access to:
  - IMS/VS physical terminals.
  - IMS/VS transactions.
  - IMS/VS control regions.
  - Control region resources (PSBs, transactions, and logical terminal names) for message processing regions and batch message processing regions.
- Providing an audit trail to the individual operator on the IMS/VS log

For more detail on using RACF to protect IMS/VS data sets and other resources, refer to *RACF Security Administrator's Guide* and the IMS/VS library.

4.9.3 RACF and TSO/E

On MVS systems with TSO/E Version 1 Release 4 or later installed, you can use RACF to protect TSO/E resources. RACF provides the following classes for protecting TSO/E resources:

TSOPROC protects TSO/E logon procedures

ACCTNUM protects TSO/E account numbers

TSOAUTH protects TSO/E user attributes OPER, JCL, ACCT, MOUNT, RECOVER, PARMLIB, TESTAUTH, and CONSOLE

PERFGRP protects TSO/E performance groups

Note that, if you are defining TSO segments in user profiles, you have to protect these resources by activating the specific class. For example, if you do not activate the TSOPROC and ACCTNUM classes, users whose profiles contain TSO segments will not be able to log on to TSO/E.

You should issue a SETROPTS RACLIST command for these TSO/E resource classes to enhance the performance.

Refer to Section 2.5, "Defining TSO Users to RACF" on page 19 for more information about logging on to the system, and to *RACF Security Administrator's Guide* for details about using RACF to protect TSO/E resources.

Chapter 5. Options

This chapter describes some of the options that you have to customize in your RACF MVS system. The SETROPTS command is used to set system-wide RACF options related to resource protection dynamically.

5.1 SETROPTS Command

When you implement RACF, select the RACF options that satisfy your security requirements. When you install RACF, almost all the options are inactive. Use the Set RACF Options (SETROPTS) command to set system-wide RACF options related to resource protection. Most SETROPTS command functions require you to have the Special or Auditor attributes.

These system-wide option settings can specify that RACF:

- Gather and display RACF statistics.
- Activate general resource classes (DASDVOL, TERMINAL, and so on).
- Activate global access checking.
- Activate generic profile checking.
- Activate program control (not for MVS/370).
- Activate and control erase-on-scratch (not for MVS/370).
- Activate enhanced generic naming for data sets and entries in the Global Access Checking table.
- Set password syntax rules, expiration interval, and activate password processing for checking previous passwords and limiting invalid password attempts.
- Set logging options. Keep in mind that each additional logging activity specified increases RACF and SMF processing and might have an impact on RACF performance.
- Refresh in-storage profile lists.
- Set the password the operator must supply in order for RACF to complete an RVARY command that changes RACF status or changes the RACF databases.
- Set installation defaults for primary and secondary national languages.
- Display options currently in effect.

Specifications in a SETROPTS command are stored in the RACF database and are valid until you change them with another SETROPTS command. That is, you do not have to issue SETROPTS at every IPL. See *RACF Command Language Reference* for the SETROPTS specifications that are in effect after the initial RACF installation and for further explanation of the SETROPTS options. To list the system-wide RACF options in effect, enter:

```
SETROPTS LIST
```

5.2 Recommended SETROPTS Options

The SETROPTS options that you will find more useful than the default values at initial installation are listed below. We assume that you have installed the MVS/DFP* product which provides the Always-Call feature.

AUDIT(*) specifies that logging will occur for all classes to record all accesses to the RACF data set by RACF command and RACDEF SVC (default NOAUDIT(*)).

GENERIC(*) activates generic profile checking for the DATASET class and all classes in the Class Descriptor table, except group-resource classes (default NOGENERIC(*)).

GENCMD(*) activates generic profile command processing for the DATASET class and all classes in the Class Descriptor table, except group-resource classes (default NOGENCMD(*)).

GLOBAL(*) specifies that the DATASET class, and all the classes in the Class Descriptor table, except group-resource classes, are eligible for global access checking (default NOGLOBAL(*)). See Section 6.1, "Global Access Checking Table" on page 59.

PASSWORD specifies a number of suboperands that monitor and check passwords. Use these options to control passwords:

HISTORY(number-previous-passwords) specifies the number of previous passwords (1 to 32) to save for each user ID and compare with an intended new password. If there is a match, the new password is rejected. The recommended number is 32 (the maximum), from a security point of view (default NOHISTORY).

REVOKE(number-invalid-passwords) specifies the number of consecutive invalid password attempts before the user ID is revoked. The recommended number is 5 (default NOREVOKE).

INTERVAL(password-change-interval) specifies the number of days (1 to 254) that a password is valid. The default password-change-interval is 30. We recommend 60 rather than 30.

RULEn(LENGTH(m1:m2) content-keyword(position)) specifies length and syntax rules for new passwords. For the first RACF installation, we recommend a length of six to eight characters and any combination of alphabetic and numeric characters (default NORULES).

WARNING(day-before-password-expires) specifies the number of days before a password expires that RACF is to issue a warning message to a user. The recommended time is 8 days (default NOWARNING).

GRPLIST specifies that access checking is based on the highest authority of the groups to which the user is connected (default NOGRPLIST).

NOADSP cancels automatic RACF protection for users with Automatic Data Set Protection (ADSP). You have to specify NOADSP when you use generic profiles because ADSP forces the creation of discrete profiles by an ADSP attribute user (default ADSP).

JES(EARLYVERIFY) specifies that JES is to invoke the System Authorization Facility (SAF). SAF calls installation-written exit ICHRTX00 (if installed) for verification of user ID, group ID, and password at job submission time. A sample of this exit is provided in Section 6.6.4, “Sample Exit - MVS Router Exit ICHRTX00” on page 71. This option is irrelevant if using JES 3.1.3 or later (default JES(NOEARLYVERIFY)).

ERASE specifies that the DASD data set extent is physically erased when it is deleted or scratched. We recommend specifying ERASE without any operand, so that a scratched data set is erased only when it has the erase indicator in the data set profile. Because erase-on-scratch places an additional load on DASDs, it can have an impact on MVS system performance, depending on how much erasure is being performed (default NOERASE). This option is ignored on MVS/370.

OPERAUDIT specifies that RACF is to log all actions allowed only because of the Operations or Group-operations attributes (default NOOPERAUDIT).

WHEN(PROGRAM) activates program control (default NOWHEN(PROGRAM)). This option is ignored on MVS/370.

RVARYPW specifies the password(s) that the operator is to use to respond to requests to approve RVARY command processing. The SWITCH and STATUS passwords are both defaulted to YES when RACF is first initialized.

LIST lists current RACF options.

5.3 AUDITOR Operands

Some SETROPTS options can be changed only by users with Auditor authority. These options are:

- AUDIT/NOAUDIT - specifies the names of the classes for which RACF is to perform auditing.
- SAUDIT/NOSAUDIT - specifies whether RACF is to log all RACF commands issued by users with Special or Group-special authority.
- OPERAUDIT/NOOPERAUDIT - specifies whether RACF is to log all actions allowed only because a user has the Operations or Group-operations authority.
- SECLABELAUDIT/NOSECLABELAUDIT - specifies whether RACF is to audit resources accessed by SECLABEL.
- SECLEVELAUDIT/NOSECLEVELAUDIT - specifies whether RACF is to audit access attempts to all RACF protected resources based on the specified installation defined security level.
- LOGOPTIONS - audits access attempts to resources in specified class based on the auditing level specified.
- CMDVIOL/NOCMDVIOL - specifies whether RACF is to log violations detected by RACF commands.

5.4 Statistics Gathering

An installation can record two types of RACF statistics. One records user logon information, and is controlled by INITSTATS; the other records access to resources in specifics classes that are protected by discrete profiles, and is controlled by STATISTICS.

When the RACF database is initialized, the default is INITSTATS on. INITSTATS is recommended because it allows you to use other options to provide additional security at logon. INITSTATS allows you to take advantage of the SETROPTS INACTIVE option and the REVOKE, HISTORY, and WARNING options of SETROPTS PASSWORD. INITSTATS records the following:

- The date and time RACF processes a RACINIT.
- The number of RACINITs for a user to a particular group.
- The date and time of the last RACINIT for a user to a particular group.

Section 1.2.1, “Database Name Table” on page 5 have some information about using INITSTATS in the primary RACF database.

When the RACF database is initialized, the default is STATISTICS off for all classes. It is recommended that you keep STATISTICS off until your installation has had an opportunity to evaluate the need for the information provided by STATISTICS versus the potential impact on performance. It is recommended that you do not record STATISTICS data on your backup database unless you use generic profiles or use SETR RACLIST to bring profiles into storage. If you use discrete profiles and do not RACLIST them, system performance may decrease sharply.

Note that statistics are not recorded for generic profiles and for in-storage profiles (in classes for which the SETROPTS RACLIST command has been issued).

5.5 Additional Considerations for SETROPTS Options

You must be careful when you specify SETROPTS options, for SETROPTS controls system-wide resource access. The following suggestions will help you avoid common mistakes at initial installation:

- DO NOT activate all classes by issuing SETROPTS CLASSACT(*). Do not use this command until you have defined appropriate profiles in the classes you plan to activate, then activate those classes only.
- DO NOT activate the SECLABEL class unless you have assigned appropriate security labels to appropriate users and to the resources that they must access.
- DO NOT activate the TERMINAL class with UACC=NONE unless you define every terminal to RACF. If you do activate it, you cannot log on from undefined terminals.
- DO NOT specify JES BATCHALLRACF or JES XBMALLRACF unless you have defined all users to RACF. If you do activate it, users who are not defined to RACF cannot submit batch jobs or execute batch monitor jobs.

- DO NOT activate PROTECTALL unless you have defined all resources in your system to RACF. If you do activate it, RACF rejects any request to create or access a data set that is not RACF-protected.
- Make sure to specify the correct date at IPL time if you use INACTIVE(unused-userid-interval) option to revoke a user who has not logged on for a long time.

After you set options for your installation, use SETROPTS LIST to display the current options and check that they are set as you specified.

If you create a new generic profile after activating generics for a class, you should issue SETROPTS GENERIC(class_name) REFRESH after adding the profile. If you do not issue the refresh, the new profile is not used by active jobs, started tasks, or TSO users currently logged on. As an alternative to a SETROPTS GENERIC(classname) REFRESH, you can issue either

```
LISTDSD DA(′dataset-name′) GENERIC
```

or

```
RLIST class-name resource-name
```

The above commands refresh data sets or general resource profiles, respectively, within the issuing address space.

For a sample jobstream to specify the SETROPTS options discussed previously in this chapter, submit job RACSETR. This job is listed in Section A.14, “SETROPTS” on page 147 and in IPO1.JCLLIB.

Note: You must have both the Special and Auditor authority to submit this job. The security administrator has the Special authority. To add the Auditor authority, issue the RACF command:

```
ALTUSER userid AUDITOR
```

The specifications provided in job RACSETR are to get you started. You will probably want to set additional options based on to your installation′s environment.

5.6 Using SETROPTS RACLIST and SETROPTS GENLIST

The RACLIST function on the SETROPTS command improves performance by copying generic and discrete profiles for the general-resource class from the RACF database into storage. Since RACLIST minimizes I/O, its use is recommended.

Before you use RACLIST, consider how frequently the class is referenced, the number of profiles in the class, and the amount of storage that would be required to hold the profiles. Use SETROPTS RACLIST when the general resource class contains a small number of frequently referenced profiles, and global access checking cannot be used. See *System Programming Library: RACF* for details about storage estimates.

The GENLIST function on the SETROPTS command improves performance by copying generic profiles from the RACF database into storage. Use SETROPTS GENLIST when the class contains a small number of frequently referenced generic profiles.

You cannot use both RACLIST and GENLIST for the same general resource class. An installation can optimize performance by carefully deciding whether to use RACLIST or GENLIST for various classes.

Before issuing a SETROPTS GENLIST or SETROPTS RACLIST for a general resource, consider whether you can afford:

- The storage used.
- The overhead. An administrator must refresh all profile changes so that they become effective.

If you create, delete, or modify a profile in a class for which:

```
SETROPTS RACLIST(class-name)
```

was issued, you must next issue:

```
SETROPTS RACLIST(class-name) REFRESH
```

to activate this new profile. This is very important for the classes in RACF 1.9 that have RACLIST REQUIRED in the Class Descriptor table.

After making changes to the in-storage profiles, you must issue a SETROPTS REFRESH command for those changes to take effect.

A REFRESH for the RACLIST class is not propagated to other systems. The REFRESH must be issued separately for each system. SETROPTS RACLIST takes effect only on a system sharing the database following a re-IPL.

Some RACF classes are RACLIST required. If RACLIST is not invoked for these classes, no access is allowed to the resource, even if the resource is defined in the database. Although not required, DLFCLASS should also be included in RACLIST. The RACLIST required classes are:

- DEVICES
- NODES
- OPERCMDS
- PROPCNTL
- PSFMPL
- RACFVARS
- SECLABEL
- VTAMAPPL
- APPCTP
- CSFSERV
- CSFKEYS

Chapter 6. Tables and Exits

This chapter describes RACF tables, such as the Global Access Checking table, the Started Procedures table, and the Dynamic Parse table, and some installation exits.

6.1 Global Access Checking Table

The Global Access Checking (GAC) is a table built in storage and contains the names of resources that are shared among all users, and are frequently accessed by users in a common manner. When determining a user′s authority to access a resource, RACF consults the GAC table before it looks at any other security profiles. Use of this table results in less I/O to the RACF database and better system performance.

The GAC can only allow access, not deny, to a resource. If a user wishes to access a data set with a higher level than specified in the GAC or the resource is not found in the table, RACF profile checking is performed. The Global Access Checking Table is strictly a performance and tuning tool.

Note: When RACF grants access to a resource because of an entry in the GAC table, RACF does not log the event (even if you request logging) and maintains no statistics.

6.1.1 Special Qualifiers

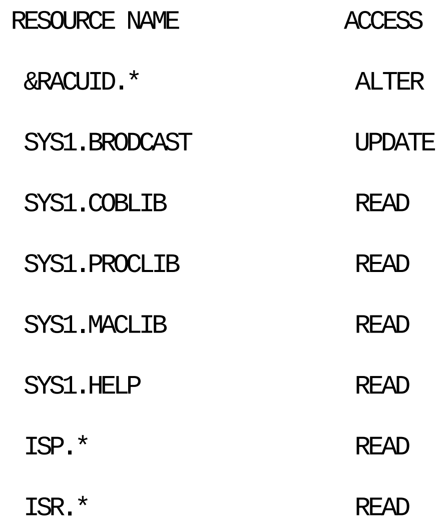
There are two special qualifiers that are used only in the global access checking table:

- &RACUID - Allows users the specified access to data sets that begin with their own user IDs. It speeds the process by which RACF grants the access and also prevents auditing of access attempts.
- &RACGPID - Gives all users the specified access authority to data sets whose high-level qualifier is the user′s current connect group.

6.1.2 GAC Table Example

Figure 13 shows a sample GAC table. In the figure:

- If the data set begins with the RACF user ID of the user who is accessing the data set, the access is allowed with the authority of ALTER.
- Anyone can update SYS1.BRODCAST.
- Anyone can read SYS1.COBLIB.
- Anyone can read SYS1.PROCLIB.
- Anyone can read SYS1.MACLIB.
- Anyone can read SYS1.HELP.
- Anyone can read all the libraries of ISPF/PDF (ISP.* and ISR.*).



RESOURCE NAME	ACCESS
&RACUID.*	ALTER
SYS1.BRODCAST	UPDATE
SYS1.COBLIB	READ
SYS1.PROCLIB	READ
SYS1.MACLIB	READ
SYS1.HELP	READ
ISP.*	READ
ISR.*	READ

Figure 13. Global Access Checking Table

In the GAC table, you should specify the most frequently used data sets in your installation, such as:

- ISPF/PDF libraries
- SDSF libraries
- Compiler libraries (COBOL, PL/I, ASM, and FORTRAN), such as SYS1.PLIBASE
- Tool libraries
- SYS1.BRODCAST
- SYS1.HELP
- SYS1.PROCLIB

Additionally, you may want to specify other entries for the GAC table, using the following guidelines:

- Attempt to identify resource profiles that are accessed frequently and for which a performance benefit is desired.
- Do not add entries to the GAC table for profiles in classes that are in the RACLIST. RACF will not search the GAC table for profiles in classes that are in RACLIST.
- For resource profiles with a UACC other than NONE, consider adding similar entries to the GAC table.
- For resources protected by a generic profile with a UACC other than NONE, and other resources protected by a profile that has specific access requirements such as an access list, consider adding two entries: one for the larger set of resources (with UACC equal to the UACC of the profile) and the other for the smaller set of resources (with access authority of NONE). For example, if you have a profile of SYS1.** with a UACC(READ), but you also have some specific profiles with more restrictive entries, such as SYS1.PARMLIB with UACC(NONE), then create two entries:

```
SYS1.PARMLIB/NONE
SYS1.**/READ
```

Do not add a GAC table entry if any of the following conditions exist:

- The profile has a security level, security category, or security label other than SYSLOW. If the profile has a security label of SYSLOW, the GAC table entry can have an access of READ.
- The profile has an entry in the standard access list that is lower than the access level of the GAC table entry.
- The profile has an entry in the conditional access list that is more restrictive than the access level of the GAC table entry.
- The profile requests auditing of successful access attempts at or below the level specified in the corresponding GAC table entry.

6.1.3 GAC Table Maintenance

The GAC table is maintained with the general resource maintenance commands for a class of resources called GLOBAL. These commands are shown in the GAC table definition in Figure 14. When you want to change the GAC table, use the RALTER command. You can list the GAC table using the RLIST command. For every change made in the GAC, you must refresh storage.

When you define an entry for a data set in the GAC table, enclose the entry name in quotes if you do not want your TSO prefix (which might be your user id) used as the high-level qualifier of the entry name.

```
 RDEFINE GLOBAL DATASET ADDMEM(′&RACUID.*′/ALTER)
   RALTER GLOBAL DATASET ADDMEM(′SYS1.BRODCAST′/UPDATE)
   RALTER GLOBAL DATASET ADDMEM(′SYS1.COBLIB′/READ)
   RALTER GLOBAL DATASET ADDMEM(′SYS1.MACLIB′/READ)
   RALTER GLOBAL DATASET ADDMEM(′SYS1.PROCLIB′/READ)
   RALTER GLOBAL DATASET ADDMEM(′SYS1.HELP′/READ)
   RALTER GLOBAL DATASET ADDMEM(′ISP.*′/READ)
   RALTER GLOBAL DATASET ADDMEM(′ISR.*′/READ)
  RLIST GLOBAL *
```

Figure 14. Global Access Table Definition

You can use the SETROPTS command to activate and deactivate global access checking and to refresh the GAC table in storage as follows:

- To activate:

  ```
  SETROPTS GLOBAL(DATASET)
  ```

  If you specify GLOBAL(*) you activate global access checking for the DATASET class and all classes in the Class Descriptor table except group resource classes.

- To deactivate:

  ```
  SETROPTS NOGLOBAL(DATASET)
  ```

- To refresh:

  ```
  SETROPTS GLOBAL(DATASET) REFRESH
  ```

To execute these commands, you must have Special authority. For further details, refer to *RACF Command Language Reference*.

For the JCL to create the GAC table sample shown in this section, refer to job RACFGAC. The SETROPTS GLOBAL(DATASET) command, which activates these changes in the GAC table, is issued at the end of job RACFGAC. This job is listed in section A.15, “Global Access Checking Table” on page 148 and can be found in IPO1.JCLLIB.

6.1.4 Additional Considerations

When you use global access checking, consider the following:

- The group-resource classes (such as GTERMINL) are ineligible for global access checking.
- When global access checking allows a request, RACF maintains no statistics.
- When global access checking allows a request, RACF performs no logging other than that requested by the SETROPTS LOGOPTIONS command.
- Update of GAC table entries become effective at the next IPL or after execution of the SETROPTS command with the GLOBAL(class-name) operand (with or without the REFRESH operand).
- While RACF is searching the GAC table for a matching entry, profiles in the class are ignored. If no GAC table entry matches the search, or if the access specified in the entry is less than the access being requested, RACF then searches for a matching profile in the class.
- The only use for an access of NONE in the GAC table is to force RACF to look for a profile. This could be used when you have access list entries that have a lower access level than a data set′s UACC, or when you want to ensure that auditing or security classification checking will take place for a specific data set.
- Because RACF performs global access checking before security classification processing, global access checking might allow access to a resource you are protecting with a security category, security level, or both. To avoid a security exposure to a sensitive resource, do not define an entry in the GAC table for a resource you are protecting with security classification processing.
- Save a listing of the GAC table. This listing can assist you in recovering from the accidental deletion or alteration of the GAC table or its entries. You can use the RLIST command to make this listing quickly.
- It is recommended that, for each entry in the GAC table, you create a similar resource profile. Such a “matched pair” approach can help ensure the continuation of protection if the GAC table becomes disabled. For example:

  ```
  RDEFINE class resource UACC(access)
  ```

6.2 Class Descriptor Table

The Class Descriptor table (CDT) describes attributes of general resource classes. RACF is, with the exception of user, group, and data set profiles, strictly table-driven. For each general resource class, there is a unique entry in the table. RACF references the CDT whenever it receives a resource class name other than DATASET, USER, or GROUP.

This technique allows new resource classes to be added as needed to support more resource managers and new applications, and also enables a RACF installation to define its own resource classes. If you update the CDT, you must also update the RACF Router table.

All resource classes are listed in the two load modules of the CDT:

- ICHRRCDX is the name of the IBM supplied class entries.
- ICHRRCDE is the name of the installation defined class entries.

Note: The CDT must be the same for each system sharing the RACF database. If there are different tables, unpredictable results can occur when you use the SETROPTS command to activate or deactivate RACF protection for the class.

See the *System Programming Library: RACF* for information on creating entries in the CDT.

6.3 Router Table

The RACF Router table contains one or more entries for each entry in the IBM supplied portion of the CDT. It also contains entries for DATASET and USER.

If an entry is added to the CDT, a corresponding entry must also be added to the RACF Router table, with a blank REQSTOR and blank SUBSYS field. Otherwise, you will not be able to issue the SETR RACLIST command for that class. You may also have entries in the Router table that do not appear in the CDT.

The Router table consists of two load modules:

- ICHRFR0X is the name of the IBM supplied Router table.
- ICHRFR01 is the name of the installation supplied Router table.

Note: Do not delete or modify any entries in ICHRFR0X. Doing so can result unpredictable results.

See the *System Programming Library: RACF* for information on creating entries in the Router Table.

6.4 Started Procedures Table

Only RACF-defined users and groups can be specifically authorized to access RACF-protected resources. However, started procedures have system-generated JOB statements that do not contain the USER or GROUP parameter. Because there is no RACF user ID and group associated with the started procedure, the started procedure cannot access any RACF protected resource unless the universal access of the resource allows access. To avoid this situation, RACF uses a table to associate a user ID and a group name with a started procedure. This table is the Started Procedures table (SPT), which resides in module ICHRIN03 in the SYS1.LPALIB library. With this table, normal access authorization checking is performed for a started procedure using the associated RACF user ID and group name.

Some started procedures need special authorization to access RACF protected resources. For example, you can allow JES to access SPOOL data sets, which can be authorized in two ways:

- When the Bypass Password Protection Bit is on in the Program Properties Table (PPT), RACF is not called for OPEN, EOV, or DADSM rename or scratch. This bit is associated with the program name.
- When the PRIVILEGED or TRUSTED bit is set on in the SPT, RACF does no access authorization checking unless the RACHECK macro was specified with ENTITY=(...,CSA).

You should use the PRIVILEGED or TRUSTED bit in the RACF SPT rather than the “bypass password protection bit” in the PPT to authorize started procedures to bypass RACF authorization checking because it is easier to maintain security controls in only one table. Also, RACF then controls which users can bypass RACF.

The PRIVILEGED indicator, available in earlier RACF releases, allowed RACF access without logging for almost all RACHECKs. PRIVILEGED indicator in RACF 1.9 provides more privileges than the PPT NOPASS indicator since it grants access in all classes; NOPASS grants access to only data sets. RACF 1.9 introduces the SPT TRUSTED attribute, which grants the same access as the PRIVILEGED attribute, but provides the ability to log the access. Logging on SMF is actually controlled by the SETROPTS LOGOPTIONS, rather than by individual RACF profiles. The PRIVILEGED attribute is still provided by RACF 1.9 and if a started procedure has both the PRIVILEGED and the TRUSTED attributes, the PRIVILEGED attribute overrides and disallows logging. TRUSTED means the same as PRIVILEGED, except that auditing can be requested by using the SETROPTS LOGOPTIONS command.

TRUSTED is useful in setting up the proper operation for started tasks such as JES2, JES3, PSF, VTAM, LLA, VLF, SMF, MOUNT, DUMPSRV, and CATALOG.

There can be only one generic entry in the SPT and it must be the last entry. Characters of * and = will not be acceptable in other entries.

As with any other user ID and group name, the user ID and group name that you assign to a started procedure must be defined to RACF using the ADDUSER and the ADDGROUP commands. You may also have to authorize the users or groups to the required resources using the PERMIT command. If a started procedure is executed without associating its name with a RACF defined user ID and group name, the started procedure runs as an undefined user and can access RACF-protected resources if the UACC for the resource is sufficient to allow the requested operation. No user verification (password checking) takes place for a started-procedure user ID.

Neither TRUSTED nor PRIVILEGED will propagate; they are given only to started tasks. If a started task opens the internal reader and puts a job into the system, the job will run under the started-task user ID if there is no USER= on the job card. However, the job will be neither TRUSTED nor PRIVILEGED. This may be important to think about if you run a scheduling package.

The IBM default SPT does not contain any entries. It must be set up by the installation by coding an Assembler language module and link-editing it as ICHRIN03 in SYS1.LPALIB or the equivalent. Good control over the SPT attributes requires that write access to all procedure libraries and all load libraries is well controlled.

For more details about the coding the ICHRIN03, see *System Programming Library: RACF*.

The started procedure names, the associated user ID and group name, and the PRIVILEGED or TRUSTED bit setting in the SPT can be listed using DSMON. For information on how to run DSMON, refer to section 8.2, “Data Security Monitor” on page 86.

6.4.1 Started Procedures Table Example

The SPT in Figure 15 contains four entries:

- The first two entries represent the different JES program products. Because these subsystems must have special access to the SPOOL data sets, the TRUSTED bit is set on for these procedures.
- The third entry in the sample table is for IRRDPTAB in order to load the RACF Dynamic Parse tables. The user ID STCDPT is neither PRIVILEGED nor TRUSTED.
- The last entry in the started procedures table is a generic entry specified by an asterisk (*) in the procedure name field. This generic entry is used for any started procedures that do not match any of the previous entries in the table. An equal sign (=) is specified in the user ID field to indicate that the procedure name will be used as the user ID. The user ID should not have any special authorization.

The group STCGRP is used for the group name for all started procedures. The group structure and the group STCGRP are discussed in section 2.2.2, “System Management Group Structure” on page 15. For administrative and security reasons, it is advantageous to define a separate user ID for started procedures that require any special authorization, such as OPERATIONS.

PROCEDURE	USERID	GROUPID	PRIVILEGED	TRUSTED
JES2	STCJES	STCGRP	NO	YES
JES3	STCJES	STCGRP	NO	YES
IRRDPTAB	STCDPT	STCGRP	NO	NO
*	=	STCGRP	NO	NO

Figure 15. Started Procedures Table

6.4.2 Installation

If a procedure name changes, the corresponding table entry must be changed. Also, additional procedures can be added to the SPT.

For the JCL to install the sample SPT refer to job RACSTC on A.16, “Started Procedures Table” on page 149 and in IPO.JCLLIB. In the first step of this job, the user IDs STCJES and STCDPT are defined to RACF. You may have to authorize these users to the required resources using the PERMIT command. The group STCGRP should have previously been defined with the job RACGRP, described in section 2.2.2, “System Management Group Structure” on page 15. The next step of the job executes only if the first step ends with a return code of zero. The SPT is then assembled and placed in the SYS1.LPALIB library using SMP/E.

After replacing the ICHRIN03 module, you must re-IPL the system with the CLPA option for the new module to be in effect.

Note: You may receive a return code of 8 in STEP INSTL3 if entry ICHRIN03 already exists in SYS1.LPALIB. This can be ignored.

If you are migrating RACF to the Release 1.9, you must update the SPT (ICHRIN03) to include these trusted procedures:

- JES2 or JES3
- MOUNT
- PSF
- VTAM
- LLA
- VLF
- SMF
- CATALOG
- DUMPSRV
- DLF

6.5 Dynamic Parse Table

The RACF command processors require a dynamic table to be built after each IPL to parse segment related keywords. This table is used when dealing with profile segments to add, list, alter, or delete DFP, TSO, or any other non-base segment information with the RACF commands. The table is built by the IRRDPI00 module.

Commands that refer only to RACF base information will work, provided they do not contain any typing errors. If you make a mistake, dynamic parse will parse the command and issue a message for an invalid entry (the typing error). Commands the administrator enters that reference the profile segments will fail if dynamic parse is not installed and activated.

You can automatically invoke the IRRDPI00 UPDATE command through a started task which must execute after every IPL. The steps are documented in *RACF Program Directory for MVS Installations*. Some of the necessary steps have been completed for you in the CBIPO:

- CBIPO has added IRRDPI00 to the TSO/E APF authorized command and program tables.
- CBIPO has added a procedure called IRRDPTAB to SYS1.PROCLIB.
- Procedure IRRDPTAB is a started task that runs at IPL to load the Dynamic Parse table by way of the IRRDPI00 UPDATE command. The IRRDPTAB procedure must be added to the RACF SPT. This will have been completed if you have run job RACSTC as described in section 6.4, “Started Procedures Table” on page 63.

Note: Customers are not allowed to change the dynamic parse specification data. The RACF Dynamic Parse Specification Data Set resides in SYS1.SAMPLIB, member name IRRDPSDS. You should consider maintaining an offline copy of the IRRDPSDS, moving it to another data set, and making sure to prevent anyone from modifying its contents. Put IRRDPSDS in a data set that you are sure will be part of your disaster and recover system volume set.

RACF authorization of the IRRDPI00 command has NOT been completed by your CBIPO. The *RACF Program Directory for MVS installations* indicates that the user invoking the IRRDPI00 module can be permitted to IRRDPI00 in the FACILITY class:

```
SETROPTS CLASSACT(FACILITY)
RDEFINE FACILITY IRRDPI00 UACC(NONE)
PERMIT IRRDPI00 CLASS(FACILITY) ID(STCDPT) ACCESS(READ)
```

You can, instead of using the FACILITY class, define IRRDPI00 as a program control and give READ access to the user who executes the program.

Use of the STCDPT user ID in the PERMIT command assumes that job RACSTC has been run. The actual start command for the IRRDPTAB procedure has NOT been included in the CBIPO. The COMMND00 member of SYS1.PARMLIB must be updated to include a START command for the IRRDPTAB procedure. Add the following line to the COMMND00 member:

```
COM='START IRRDPTAB'
```

Note: The Dynamic Parse Table does not function with TSO/E 1.3.0, which is the release that is available in the CBIPO 370 systems. Within the CBIPO 370 systems, the IRRDPTAB procedure has not been included in SYS1.PROCLIB.

6.6 Installation Exits

The installation exits are product-sensitive programming interfaces. RACF provides a number of exits to meet your special security requirements and let you use your own routines to enhance the facilities offered by IBM, as well as to optimize their usability. This section helps you find which exits you should code to fulfill your requirements.

The major functions within RACF have exits. There are exits for:

- User identification and verification
- Access checking
- Automatic profile creation and deletion
- Resource-, user-, and group-profile manipulation commands

RACF exits can be divided into two types, SVC related exits and other exits. Each type is discussed in the following sections.

For questions related to RACF programming interfaces, refer to *System Programming Library: RACF*.

6.6.1 SVC Related Exits

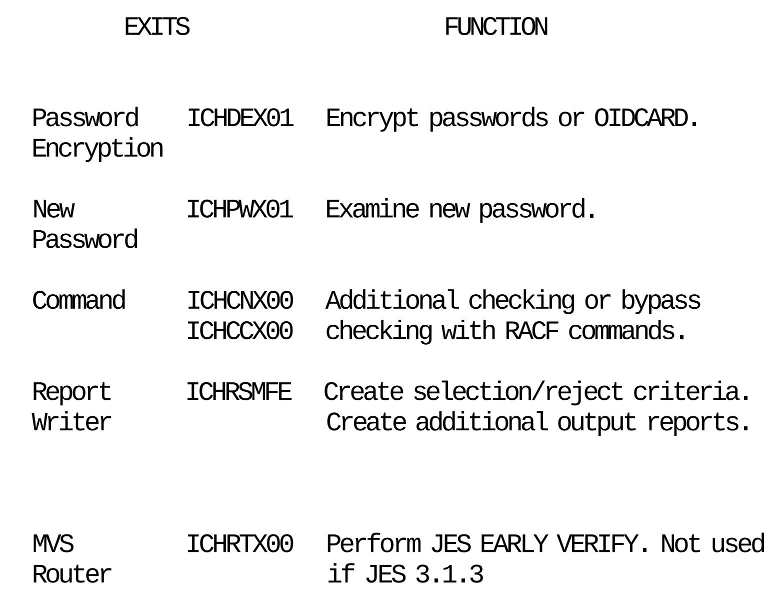
SVC related exits gain control before and after RACF SVC processing. Figure 16 lists the RACF macros used to invoke the RACF SVCs, the functions of the RACF SVC, and the RACF exit names. The FRACHECK macro is included in this list even though it does not invoke an SVC because its function is similar to that of the RACHECK macro.

MACRO	FUNCTION	EXITS	TYPE
RACDEF	Define or delete profiles.	ICHRDX01 ICHRDX02	pre-processing exit postprocessing exit
RACHECK	Determine if the user is authorized to the resource.	ICHRCX01 ICHRCX02	pre-processing exit postprocessing exit
RACINIT	Determine if the user ID and password are correct.	ICHRIX01 ICHRIX02	pre-processing exit post-processing exit
RACLIST	Build in-storage copies of general resource profiles.	ICHRLX01 ICHRLX02	pre- or postproc. exit selection exit
FRACHECK	Check authorization to resources protected by RACLIST-created profiles.	ICHRFX01 ICHRFX02	pre-processing exit postprocessing exit

Figure 16. RACF SVC Related Exits

6.6.2 Other Exits

The other RACF exits gain control at various points in RACF processing, such as during RACF TSO command processing. Figure 17 lists these RACF exits, their names, and their functions.



EXITS		FUNCTION
Password Encryption	ICHDEX01	Encrypt passwords or OIDCARD.
New Password	ICHPWX01	Examine new password.
Command	ICHCNX00 ICHCCX00	Additional checking or bypass checking with RACF commands.
Report Writer	ICHRSMFE	Create selection/reject criteria. Create additional output reports.
MVS Router	ICHRTX00	Perform JES EARLY VERIFY. Not used if JES 3.1.3

Figure 17. Other RACF Exits

You can modify the exits to perform some of the following functions:

- Allow access to a RACF-protected resource when RACF is inactive
- Enforce rules for the contents of passwords
- Protect the user from accidental destruction of data
- Modify data-set naming conventions in MVS
- Ensure tape protection for tapes without a profile in MVS

For more details about exit modification, see *System Programming Library: RACF*.

The Data Security Monitor (DSMON) produces the RACF Exits report. This report list the names of all the installation-defined RACF exit routines and specifies the size of each exit routine module. You can use the information in this report to verify that only those exit routines that have been defined by your installation are active. The existence of any other exit routines might indicate a system security exposure, because RACF exit routines can be used to bypass RACF security checking. Similarly, if the length of an installation defined exit routine module differs from the length of the module when it was defined by your installation, you should notify your RACF security administrator, because the module might have unauthorized modifications. For more details about DSMON, refer to section 8.2, “Data Security Monitor” on page 86.

6.6.3 How Exits Get Control

Figure 18 shows some MVS and TSO functions that invoke RACF functions, and give RACF exits control. With this information, an installation can determine whether RACF exits could perform extra security functions. For example, the TSO LOGON process issues an MVS RACROUTE REQUEST=VERIFY macro, which invokes the RACF RACINIT function. RACF exit ICHRIX01 gets control, RACINIT processing occurs, RACF exit ICHRIX02 gets control, and RACINIT returns control to the TSO LOGON process. An installation could use these RACINIT exits to perform extra security checks at TSO LOGON time.

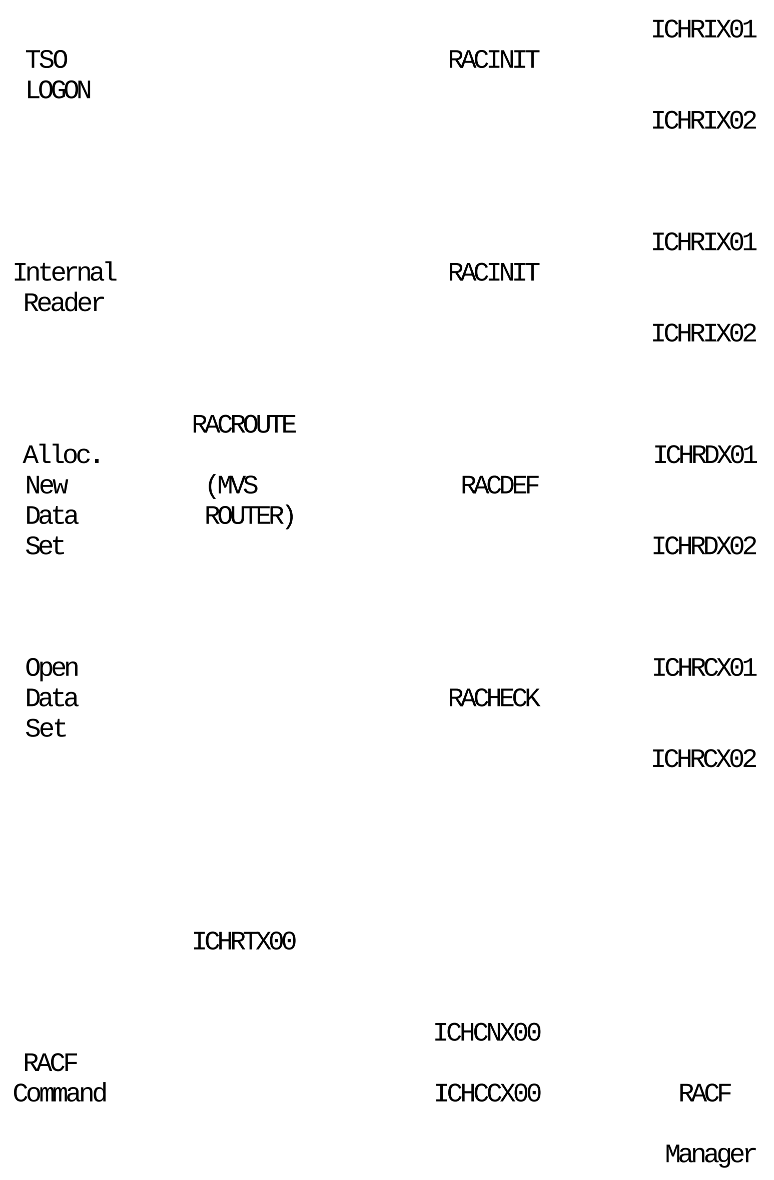


Figure 18. How RACF Exits Get Control

6.6.4 Sample Exit - MVS Router Exit ICHRTX00

With the current release of RACF, installation exits are no longer needed for a secure MVS system. Generally, exits should be kept to a minimum. The sample exit included here improves RACF processing in a specific area. The following description of the exit′s function will permit you determine whether it would be useful to your installation.

This exit provides extra security checking at RACROUTE processing time. The exit is not needed for RACF 1.9 with JES 3.1.3. Installation ICHRTX00 is not a RACF exit, although, in a system with RACF installed, it is used for RACF-related tasks. The router (or System Authorization Facility), as the interface between RACF and the resource managers, is an MVS component that is in the system, regardless of whether RACF is installed or not. For information about the System Authorization Facility (SAF), refer to Section 10.3, “System Authorization Facility Interface” on page 128.

MVS ICHRTX00 is entered for each invocation of the MVS router (through the RACROUTE macro). Therefore, the exit code must determine the types of invocations it is going to handle, and return directly in all other cases. Since there is only one router exit, exit versions that handle different router invocations have to be carefully combined into one module. The rest of this section discusses the sample exit provided in job RACRTX0. You must have Assembler H (5668-962) on your system to run this sample exit.

Listing for exit ICHRTX00 is in Section A.17, “Router Exit” on page 151.

6.6.4.1 Purpose

When JES early job verification is activated with the SETROPTS EARLYVERIFY command, JES2 1.3.4 and JES3 1.3.4 (after the appropriate PTFs have been installed) issues the RACROUTE REQUEST=VERIFY,ENVIR=VERIFY macro. This router call is not provided by RACF, and ICHRTX00 is needed to verify the user ID and password. If a user, or a job scheduling system such as Operations Planning and Control (OPC*) or Secondary Operator Facility (SOF), submits jobs with a user ID different from its own, it is sometimes desirable to have an authorization mechanism that allows these users to submit jobs for designated user IDs without having to specify passwords. This approach has been implemented in this sample exit with what is called “surrogate user support.”

Note: The MVS router exit is not invoked for all jobs that enter the system. The exit is not invoked for:

- TSO logons.
- Started procedures.
- Jobs entering the system through the internal reader interface that are subject to user ID propagation.

6.6.4.2 Functions

The sample MVS router exit has two functions:

- If the parameters USER= and PASSWORD= have been specified on a JOB JCL statement, RACINIT is used to verify the user′s access to the system. The RACINIT parameter STAT=NO is used for reasons of performance and correctness of the statistical data. SMC=NO is used because otherwise, all other tasks in the JES address space would be stopped during the RACINIT. If verification fails, the exit returns with return code 208 and the RACINIT return and reason codes are stored in SAFPRRET and SAFPRREA, respectively. The job will be flushed and message IEF722I with the appropriate message text will be issued.

- If the parameter USER= has been specified on the JOB JCL statement without the PASSWORD= and NEWPASSWORD= parameters (and with STAT=NO and SMC=NO), a RACINIT without password check is performed for the submitter′s user ID. A RACHECK is then done in class FACILITY for resource $SUBMIT.userid where userid is the one specified in the USER= parameter of the submitted job′s JOB JCL statement. An access authority of READ or higher is required for the submitter. If the user lacks sufficient authorization, if a profile for $SUBMIT.userid does not exist, or if the FACILITY class is not active, the exit returns with return code 208 and X′18′ in SAFPRRET. This causes the job to be flushed with the message:

  ```
  ′IEF722I - jobname FAILED - FAILED BY INSTALLATION′
  ```

  If the submitter has sufficient authorization from profile $SUBMIT.userid, JES treats the job as though userid propagation had occurred.

6.6.4.3 Installation

The exit should be installed with the functions of SMP/E. The FACILITY class must be activated and early job verification in JES must be activated. Sample JCL and Assembler source are contained in job RACRTX0 in Section A.17, "Router Exit" on page 151 and in IPO1.JCLLIB. You must have Assembler H on your system to run this sample exit.

The following profile must be defined for any user ID that is eligible for job submission by other users through "surrogate user support":

```
RDEFINE FACILITY $SUBMIT.userid1
PERMIT $SUBMIT.userid1 CLASS(FACILITY) ID(userid2) ACCESS(READ)
```

where userid2 represents the user who is to be authorized to submit jobs with USER=userid1 on the JOB JCL statement without the PASSWORD= parameter.

For example, if user OPC should be authorized to submit jobs for user PAYROLL, the profile $SUBMIT.PAYROLL must be created and user OPC must have READ authority.

Chapter 7. Utilities

This chapter describes the RACF utilities. These programs can be used to verify user access, to unload the RACF database, and to perform maintenance functions, such as as verify, reorganize, expand, and copy, on the RACF database.

If you are using a restructured RACF database, you should use the IRRUTx00 utilities. If you are using a non-restructured RACF database you should use the ICHUTx00 utilities.

Fore more details about the use of the RACF utilities, see *System Programming Library: RACF* and *RACF Diagnosis Guide*.

Figure 19 shows the relationship between the utilities that can be used on restructured and the non-restructured RACF databases.

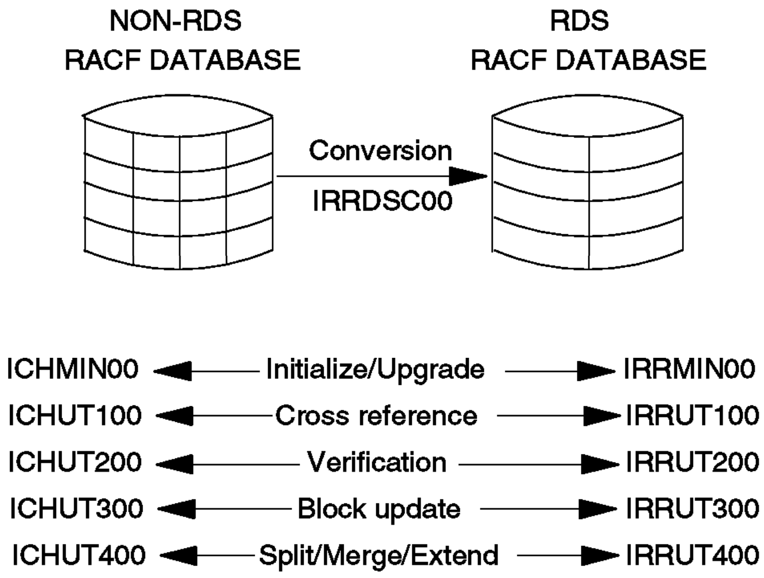


Figure 19. New RACF 1.9 Utilities

In addition, a new Database Unload Utility (IRRDBU00) can be used to unload the contents of a restructured RACF database into a sequential file for further processing.

This chapter does not describe the IRRDSC00 and IxxMIN00 utilities. For a description of these utilities, refer to Section 1.7, “Converting RACF Database to Restructured Format” on page 9 and to Section 1.1.4, “Additional Consideration for Existing RACF Installations” on page 4.

7.1 Cross-Reference Utility Program (IRRUT100/ICHUT100)

IxxUT100 lists all occurrences of a user ID or group name that are in a RACF database. To invoke IxxUT100 you must be a RACF defined user and either have the Special, Group-special, Auditor or Group-auditor authority, or be requesting a list of occurrences only for your user ID.

Figure 20 is an example of the job to run the IxxUT100 utility. In this example, IxxUT100 locates all occurrences of user IDs CARLOS and SERGIO in the RACF database and prints these occurrences on the system output device. The program name must be changed to IRRUT100 or ICHUT100, depending on your RACF database format (RDS or NRDS, respectively).

```
//JOB1     JOB
//STEP1    EXEC PGM=IxxUT100
//SYSUT1   DD   UNIT=SYSDA,SPACE=(TRK,(5,1))
//SYSPRINT DD   SYSOUT=*
//SYSIN    DD   *
   CARLOS    SERGIO
/END
```

Figure 20. ICHUT100 / IRRUT100 Example

7.2 Database Verification Utility Program (IRRUT200/ICHUT200)

IxxUT200 can be used to identify inconsistencies in the internal organization of a RACF database and also to make an exact copy of the RACF database. It performs the following functions:

- Scans the index blocks and prints information about problems with the index block chains.
- Compares the segments of the database that are actually in use to the segments allocated according to the BAM blocks, and prints information about inconsistencies.
- Creates a backup copy of a RACF database.

For RDS databases only, IRRUT200 also performs:

- Validates and reports errors found in the SEGMENT relative-byte addresses (RBAs) of the BASE segment of all profiles.
- Validates that index entries point to the correct profile.
- Validates the database format.
- Issues return codes to signal validation errors
- Creates an enhanced, formatted index report displaying the 255-byte profile name and profile type information.

Note: You cannot use IxxUT200 to enlarge your database, only make an exact copy. To enlarge your database, use IxxUT400.

For proper utility operation, the enhanced generic name (EGN) setting of the database that you are processing with IxxUT200 should be the same as the EGN setting of the system upon which the utility is being executed. Using the

IxxUT200 utility in a mixed EGN environment may cause a “?” to be displayed as a part of a formatted index entry.

The following control statements can be used:

- INDEX - specifies that you want index scan function performed.
- INDEX FORMAT - specifies a formatted listing of all the index blocks.
- MAP - specifies that you want BAM/allocation verification performed.
- MAP ALL - specifies that you want the encoded map for each BAM block in the RACF data set printed.
- END - terminates the utility program.

Figure 21 is an example of the job to run the IxxUT200 utility. In this example, IxxUT200 copies a RACF database to the SYSUT1 data set. A summary listing of all the index blocks is printed. Any block availability mask (BAM) that contains conflicts is also printed with a table of the locations of the conflicts. The program name must be changed to IRRUT200 or ICHUT200, depending on your RACF database format (RDS or NRDS, respectively).

```
//JOB1     JOB
//STEP1    EXEC PGM=IxxUT200
//SYSRACF  DD   DSN=SYS1.RACF,DISP=SHR
//SYSUT1   DD   UNIT=SYSDA,SPACE=(CYL,(10)),DCB=(LRECL=1024,RECFM=F)
//SYSUT2   DD   SYSOUT=*
//SYSPRINT DD   SYSOUT=*
//SYSIN    DD   *
  INDEX
  MAP
  END
/*
```

Figure 21. ICHUT200/IRRUT200 Example

7.3 Block Update (BLKUPD) Command Utility (IRRUT300/ICHUT300)

BLKUPD modifies the records in the RACF database. This utility is used to correct any inconsistencies that the RACF database verification utility (IxxUT200) identifies. You can examine or modify any BAM, index, or data block in a RACF database. BLKUPD runs as an authorized program.

Before you use this command, you should be very familiar with the RACF database and its configuration because improper specification can result in damage to the RACF database.

For information on how to use this utility refer to *RACF Diagnosis Guide*.

7.4 Database Split/Merge/Extend Utility Program

(IRRUT400/ICHUT400)

IxxUT400 performs the following functions:

- Copies the RACF database to a larger or smaller database.
- Splits a single RACF database into multiple RACF databases.
- Merges multiple RACF databases into fewer RACF databases.
- Rearrange RACF profiles across the same number of input and output RACF databases.
- Identifies inconsistencies, such as duplicate resource names appearing in different classes.

For RDS databases only, IRRUT400 also physically reorganizes the database by bringing all segments of a given profile together.

You should run IxxUT400 when your system has little RACF activity to reduce system performance impacts.

The following control statements can be used:

- LOCKINPUT - The RACF data set used as input cannot be updated, with the exception of updating statistics, even after the utility terminates.
- TABLE - Permits a specification of a user-written range table to be used to select an output database for each profile.
- FREESPACE - Allows you to control the amount of free space left in index blocks created for the output databases. You can specify that from 0 to 50 percent of the space within the index block is to be left free.
- ALIGN - Allows you to control profile space allocation.
- DUPDATASETS - Allows you to control the processing of DATASET entries with identical names from different input databases.

Figure 22 is an example of the job to run the IxxUT400 utility. In this example, IxxUT400 copies the profiles from a single input database to a single output database. The index blocks of the output database contain free space to allow for expansion. The program name must be changed to IRRUT400 or ICHUT400, depending on your RACF database format (RDS or NRDS, respectively).

```
//JOB1      JOB
//STEP1     EXEC PGM=IxxUT400,PARM=′FREESPACE(20)′
//SYSPRINT  DD   SYSOUT=*
//INDD1     DD   DSN=SYS1.RACF2,DISP=OLD
//OUTDD1    DD   DSN=SYS2.RACF2,DISP=(,KEEP),VOL=SER=VOL1,
//               UNIT=SYSDA,SPACE=(CYL,10,,CONTIG)
```

Figure 22. ICHUT400/IRRUT400, Copy Database Example

Figure 23 is an example of a job that split a single RACF database into two output databases. IxxUT400 assigns profiles to output databases using a range table in a module named SELECT. The load module resides in library INSTALL.LINKLIB. The program name must be changed to IRRUT400 or ICHUT400, depending on you RACF database (RDS or NRDS, respectively).

```
//JOB1     JOB
//STEP1    EXEC PGM=IxxUT400,PARM=′NOLOCKINPUT,TABLE(SELECT)′
//SYSPRINT DD   SYSOUT=*
//INDD1    DD   DSN=SYS1.RACF,DISP=OLD
//OUTDD1   DD  DSN=SYS2.RACF1,DISP=(,KEEP),VOL=SER=VOL1,
//              UNIT=SYSDA,SPACE=(CYL,5,,CONTIG)
//OUTDD2   DD  DSN=SYS2.RACF2,DISP=(,KEEP),VOL=SER=VOL2,
//              UNIT=SYSDA,SPACE=(CYL,5,,CONTIG)
//STEPLIB  DD  DSN=INSTALL.LINKLIB,DISP=SHR
```

Figure 23. ICHUT400/IRRUT400, Split Database Example

7.5 Database Unload Utility

There is a new Small Programming Enhancement (SPE) available to installations with RACF 1.9 installed. This SPE is called RACF Database Unload Utility (IRRDBU00), and allows the unloading of the contents of a restructured RACF database to a sequential (flat) file. The sequential file created can then be uploaded to a relational database management system, such as DB2, and a query manager can be used to create reports.

The information created with the IRRDBU00 utility is in a format that is particularly conductive to loading into a DB2 table, but also could be processed into ISPF tables or accessed through SLR or other standard statistical or report writing products.

Since the RACF database is a complex and encoded store for security information, the RACF database information in a relational database provide more flexibility in extracting and customizing reports.

IRRDBU00 can process an active or copy of a RACF database. We recommend that you use the utility with a copy of your RACF database as an input, since it can reduce contention to the RACF database.

The unload utility processes only a RACF restructured database. If your active RACF database is in a non-restructured format you can use the Convert Utility (IRRDSC00) to create a restructured database and then use this copy as input to IRRDBU00.

APAR OY50228 is the distribution mechanism for this utility.

7.5.1 Record Format

All data in the RACF database in unloaded, with the exception of encrypted and IBM reserved fields. Some of the fields are decoded and presented in a readable format. For example, the value of the UACC field is translated to read, update, alter, or control rather than as a binary field.

All records are identified by a 4-byte record type, with one record type per segment and per repeat group. Section Appendix B, “Records Generated by Unload Utility” on page 167 contains a table with the record types and their description.

Each record contains a name field which identifies the profile being described and permit the link with information in other records.

7.5.2 Converting and Unloading a RACF Database

This section describes the steps required for unloading the RACF database in RDS or NRDS format.

- Restructured Database (RACF 1.9 only)
  1. Copy the RACF database using IRRUT200 or use the backup database if you do not want to impact system performance
  2. Run unload utility IRRDBU00 using the copy created in previous step
- Non-restructured Database and RACF 1.9
  1. Copy the RACF database using ICHUT200 or use the backup database
  2. Define new RACF database using IRRMIN00, with PARM=NEW
  3. Convert to restructured database using IRRDSC00
  4. Run unload utility IRRDBU00 using the new restructured database
- Non-restructured Database and RACF less than 1.9
  1. Copy the RACF database using ICHUT200 or use the backup database
  2. Insert 1.9 templates in the RACF database using ICHMIN00, with PARM=UPDATE
  3. Define new RACF database using IRRMIN00, with PARM=NEW
  4. Convert to restructured database using IRRDSC00
  5. Run unload utility IRRDBU00 using the new restructured database

Note: The conversion, as well as the unload utility, must run on a RACF 1.9 system with the same classes as those in the RACF database you are processing.

7.5.3 Invoking the Unload Utility

The invocation of the unload utility is similar in operation to the dataset convert utility (IRRDSC00). If your database is split, you can process all parts or each part separately.

Some existing IRRDSC00 (convert utility) messages have been changed and may be issued by either IRRDSC00 or IRRDBU00.

Figure 24 is an example of the job to run the unload utility. INDDx represents the restructured RACF database and OUTDD represents the output flat file (LRECL=2048, RECFM=VB).

```
//JOB1     JOB
//STEP1    EXEC PGM=IRRDBU00,PARM=NOLOCK
//SYSPRINT DD   SYSOUT=*
//INDD1    DD   DSN=SYS1.RACFDB,DISP=SHR
//OUTDD    DD   DSN=SYS1.RACFDB.FLATFILE,DISP=SHR
```

Figure 24. IRRDBU00 Example

The utility supplies, as part of the SYS1.SAMPLIB, a set of DB2 tablespace, table and index definitions that define the DB2 tables to contain your unloaded database. A set of simple queries are also supplied. The members in SYS1.SAMPLIB are the following:

- RACDBUTB - Create tablespace, table, and index statements for DB2
- RACDBULD - DB load utility control statements
- RACDBUQR - Sample queries
  - Find all residual authorities in all data set access lists
  - Find all groups associated with a given user
  - Find all users associated with a given group
  - Find all users with the Special authority

7.5.4 Benefits and Uses

Using the RACF database in a flat file you can improve the system auditing extracting reports, for example to know who have privileges in the system, or what profiles exist with UACC of UPDATE, or what residual accesses exist.

You can use other data stores, such as personnel data, linked with the RACF data to produce reports, for example to know the user IDs associated with a specific individual.

Using the RACF data in a flat file can reduce the need to run the IxxUT100 utility. The advantages are that you can run a query against a copy of the database, report all IDs and not just one, and the query can be used to create commands as output.

Chapter 8. Reports

This chapter describes predefined RACF reports produced by the RACF Report Writer and the Data Security Monitor (DSMON).

In a RACF environment, different users have different levels of responsibility for security. Some people have extensive responsibility for security, while others have little or none. The primary means of defining users′ responsibility for security is the RACF user attribute. A user attribute is a part of the RACF user profile and it determines what functions a particular user is allowed to perform; that is, what authority a user has. There are a set of auditing functions that only the users with Auditor authority can perform. As auditor, you are responsible for checking that the use of RACF at an installation meets the installation′s needs for access control and accountability. Auditing access control means that you can check users′ accesses to resources and verify that the accesses allowed are appropriate to the particular resource. Auditing accountability means that you can trace activities of users on the protected system. To help you audit access control and accountability, RACF provides:

- Logging routines that record the information you require in SMF records format.
- Audit control functions that enable you to specify the logging for each resource.
- The RACF Report Writer (RACFRW), which generates user-tailored reports based on the information that RACF logged in the SMF data sets.
- The Data Security Monitor (DSMON), which generates reports containing information about the security environment of the operating system.

Examples of the jobs referred in this chapter can be found in Appendix A, “Starter System JCL” on page 133 and in IPO.JCLLIB.

For more information about the use of auditor tools (RACFRW and DSMON), refer to the *RACF: Auditor′s Guide*.

8.1 Report Writer

The RACF Report Writer (RACFRW) permits the auditor or security administrator to access the implementation of RACF and the use of the resources it protects. The report writer provides a wide range of reports that enable an installation to monitor and verify the use of systems and resources.

The report writer lists the contents of SMF records in a format that is easy to understand. It is used only with RACF SMF records. You can not use RACFRW to format other SMF records. With RACFRW, you can produce many kinds of reports, depending on your needs; for example, you can generate reports that:

- Trace access violations at LOGON time
- Trace access attempts in WARNING mode
- Trace the activity of users with Special or Operations authority
- Trace the activity of any specific user
- Trace the accesses to a specific protected resource

- Trace the use of RACF commands
- Show a general summary of activities
- Monitor the activity of all the RACF protected resources
- Describe the user and group activity

The output from the report writer includes a header page with a list of the subcommands that you have entered and describes the meanings of the event and qualifiers numbers that appear in SMF record listings and summary reports. The remainder of the report comes in various forms, depending on your request. You can request a general summary, SMF records listing, and summary reports.

Section A.18, "RACFRW Predefined Reports" on page 161 contains an example of a job to run the RACF Report Writer, and Appendix C, "RACF Report Writer Output Samples" on page 169 contains sample output reports generated by the report writer.

The RACFRW consists of three phases:

- Command and subcommand processing - interprets the report writer commands when you enter them using TSO (starting with command RACFRW) or when you run them as a batch job.
- Record selection and reformatting - compares each record from the input file (the SMF records) against the criteria you specify on the SELECT and EVENT subcommands. If you do not specify any SELECT or EVENT subcommands, the report writer selects all records from the input file. Then, records are reformatted to be sorted and printed.
- Report generation - generates the reports that you requested with the LIST and SUMMARY subcommands.

The input file to the report writer consists of SMF record types 20 (job initiation), 30 (common address work data), 80 (RACF processing), 81 (RACF initialization) and 83 (RACF processing).

To run the RACFRW as a TSO command, in the 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 preallocate the data set that contains the SMF records as RSMFIN and use it as input to the report writer command and subcommands. See *RACF Auditor's Guide* for more details about preallocating data sets on MVS.

To run the report writer at your installation, you must have:

- On MVS, IBM program product DFSORT (5740-SM1), or the equivalent.
- An output device that can handle 133-character lines.

Note:

- The RACF report writer runs as a postprocessor of RACF and does not interfere with normal RACF processing.
- Your system programmer can provide special SMF record selection and tailoring by using the RACF report writer exit routine ICHRSMFE. Fore details, see *System Programming Library: RACF*.

- In an installation using RACF to protect several systems, each MVS system writes RACF generated SMF records to a different data set. You can concatenate these data sets into a single data set for input to RACFRW. Later, you can separate them using the SYSID operand on either the LIST or the SELECT subcommand.

In IPO1.JCLLIB, there is JCL and a set of predefined RACFRW control statements that produce the reports described below. The report name is the same as the member name containing the job in IPO1.JCLLIB. An example of the JCL to list the reports is included in Section A.18, "RACFRW Predefined Reports" on page 161 and samples of output reports are included in Appendix C, "RACF Report Writer Output Samples" on page 169.

Note:

- In all the report writer sample jobs, the RSMFIN DD statement (noted by <===CHG1 in the jobs) references the IPOBAK.SMFDUMP(0) input tape Generation Data Group (GDG) data set. This input is dumped from the weekly IPOSAV.SMFDUMPW data set created using procedures identified in the SMF Procedures section of *MVS CBIPO Customization Guide*. Refer to this section for additional SMF information.

  If you submit the sample SMF jobs, that reside in IPO1.OPERLIB, from the console you must ensure that all the RACF parameters in the jobcard (USER=, GROUP=, PASSWORD=) are filled in.

- You must specify the correct VOL=SER information in the RSMFIN DD statement to reflect your installation′s SMF data tape (GDG) in the report writer jobs.

  If your installation does not use the sample SMF procedures, which utilize tape GDGs to save SMF records, change all necessary parameters in the RSMFIN DD statement in the jobs to reflect your SMF tape configuration.

8.1.1 All Activities Report - Job RACWALL

This report provides all RACF SMF status and process records, sorted by user, and a general summary report showing overall RACF-related system activity. In this report, you can trace all the activities of all users, and see when they are following the security rules and when they are not.

Note that this report can contain more than 10,000 lines, depending on the number of records that you process. We suggest that you create this report only when necessary for tracing accountability.

8.1.2 Users with SPECIAL Attribute Report - Job RACWSPE

The SPECIAL and Group-SPECIAL attributes provide the most powerful RACF authorities. You must monitor the activity of these users to ensure that they are following the security rules. This report lists the activities of all the users with the Special and Group-special attributes. You must activate the SAUDIT option of the SETROPTS command to produce the SMF records for this report.

8.1.3 Users with OPERATIONS Attribute Report - Job RACWOPE

This report includes the activities of all users with the OPERATIONS and group-OPERATIONS attributes. These attributes provide very powerful authority in a RACF environment. The users with this attribute can access almost all resources to perform operational functions such as backups. You should carefully monitor the activities of these users to ensure that all accesses to protected resources were for valid reasons. You must activate the OPERAUDIT option of the SETROPTS command before you can monitor the activity of these users.

8.1.4 Access Violations Report - Job RACWAV

This report lists access violations to any RACF protected resource and lists unsuccessful logons. RACF detects and logs any access violation. An access violation occurs when RACF denies a user access to a resource because the user is not authorized to access the resource. An access violation is a symptom that someone either misunderstood his role as a RACF user or is trying to bypass RACF protection. You can use this report to identify such users and to identify when it might be necessary to change the access list or the UACC (Universal Access Authority Code) of some protected resources. This report also produces a summary of violated resources by user and a summary of violating users by resource that can help you to identify easily the users who are violating the access rules.

8.1.5 RACF Commands Report - Job RACWCMD

The security administrator is probably the most frequent user of RACF commands. Sometimes, users without any user authority will execute ALTDSD, PERMIT, DELDSD, or similar commands against one of their own data sets. Some users, however, might 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 represents a potential security violation, and you should know about it. This report shows you every command issued to RACF, who issued that command, and whether the user was authorized to issue the command.

8.1.6 RACF Password Violation Report - Job RACWPW

This report lists password violations at logon time. This report enables you to see if the number of password violations decreases as users get accustomed to RACF, and the terminal ids on which password violations occur. You can look at the terminal identification in the listing to determine where persistent violations are originating.

8.1.7 Monitoring RACF Resources in Warning Mode Report - Job RACWWAR

This report lists accesses granted only because the Warning mode is active. When you install RACF for the first time, you can start in Warning mode; which means that RACF allows requesters to access the resource, even though the requester is not otherwise allowed access. RACF then logs these accesses in an SMF record. You can use RACFRW to monitor this access activity, and to change the RACF profiles to the correct access list and UACC values before starting normal RACF control.

8.1.8 Auditing Considerations

How long you keep the log with RACF SMF records depends on legal requirements, auditor requirements, and the possibility of problem management and trend analysis.

Options in RACF 1.9 turn on system-wide auditing and cause considerably more SMF records to be generated. Caution is advised in using SETROPTS LOGOPTION(ALWAYS(classname)).

LOGOPTIONS is the only way to audit classes for which profiles cannot be defined, such as TEMPDSN. When the TEMPDSN class is active, only the owning job step is authorized to access a temporary data set created in a job step. By setting LOGOPTIONS(FAILURES(TEMPDSN)), all failed accesses against temporary data sets can be audited.

Audit records may be created when a job is purged, and when a SYSIN/SYSOUT data sets are created, opened, selected for output, and purged. After initial testing you may wish to turn off the auditing of accesses to spool for SYSIN and SYSOUT. To do this, use SETROPTS LOGOPTIONS(NEVER(JESSPOOL)) command.

An ABEND 0C4 when running the report writer may mean that you have tried to feed reformatted SMF records from RACF 1.8 to the 1.9 RACFRW module. This is not accepted and has not been accepted in previous releases. Regular 1.8 SMF records can be used for input; it is the 1.8 report writer′s reformatting that is not accepted.

RACF 1.9 now performs SMF logging for the PROGRAM class. If you already have profiles in the PROGRAM class, examine the logging options. Logging successes in the PROGRAM class can produce a large number of SMF records. You can use SETROPTS(NEVER(PROGRAM)) to temporarily turn off logging while you examine the profiles and update the audit specifications.

There are many more calls to RACF when MVS/SP 3.1.3 and JESx 3.1.3 are installed, and more SMF records may be created even though the setting of AUDIT options in profiles is the same as previously. More SMF buffers and data sets may be needed, and a review of the applicability of the current audit options may be appropriate.

You can use the UAUDIT operand on the ALTUSER command to log all RACF-related activities for a specific user. When you set this control, RACF logs all RACF commands that the user issues; all additions, changes, or deletions that the user makes to RACF profiles; and all attempts by user to access RACF-protected resources, except those authorized by global access checking. You would probably not request user auditing as a matter of course, but it is useful if you suspect that a particular user may be misusing the system or persistently trying to access or delete resources outside the user′s control. The command is:

```
ALTUSER userid UAUDIT
```

If owner controlled logging on MVS does not provide enough information for the audit, users with Auditor authority can use the GLOBALAUDIT operand on the ALTDSD or RALTER command. With this option, the audit level that you specify in the GLOBALAUDIT command overrides the level specified in the profile in the AUDIT option. GLOBALAUDIT(NONE) is the default. After you complete the audit of the resource, it is a good practice to restore the default so that RACF logs accesses to the resource as specified in the AUDIT of the profile.

Figure 25 shows how RACF logs records in the SMF, based on the RACF options.

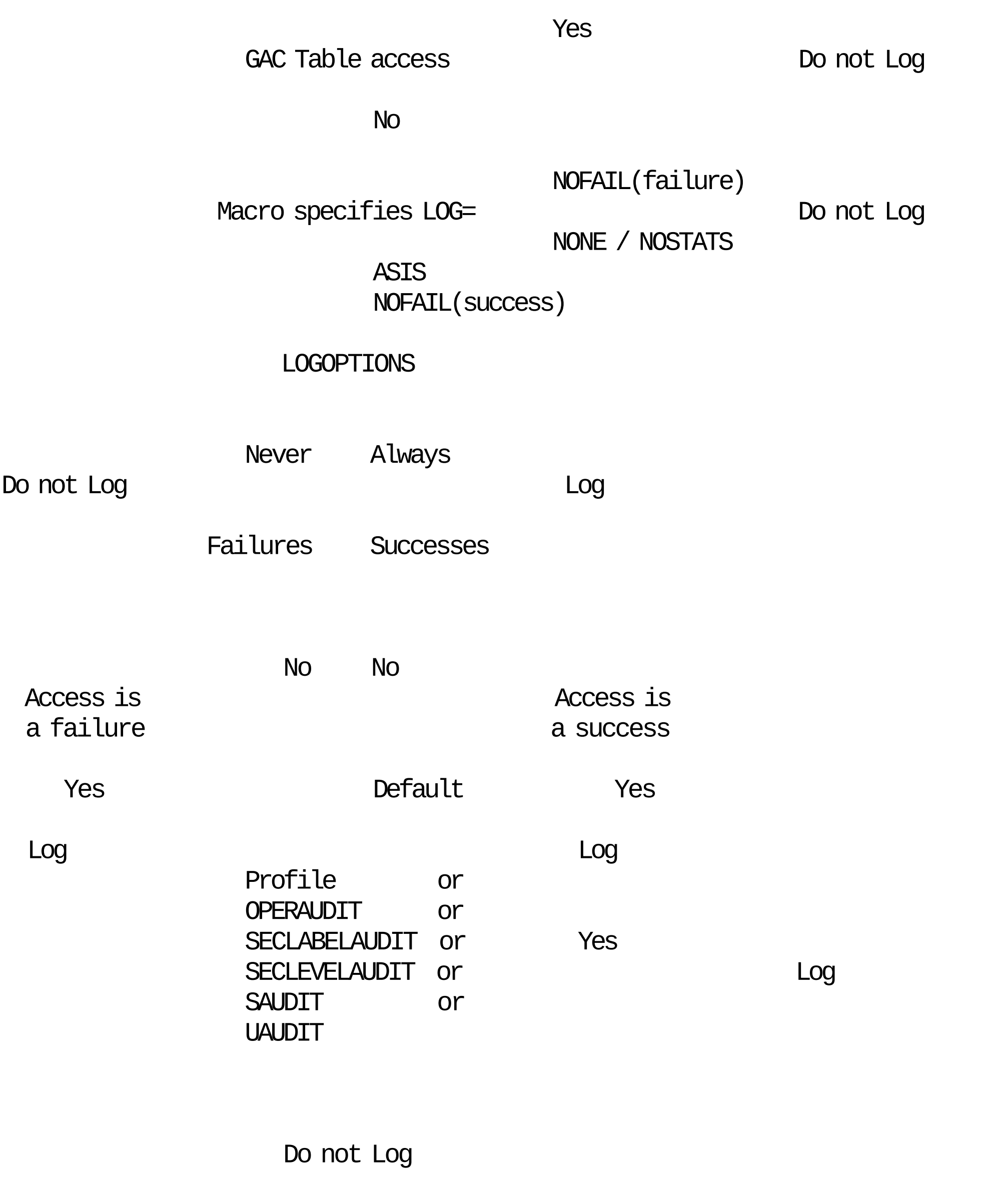


Figure 25. RACF Log Operations

8.2 Data Security Monitor

The Data Security Monitor (DSMON) program is an APF authorized batch program that can be run while RACF is active. It allows authorized users to obtain a set of reports that provide information about the current status of your installation′s data security environment and, in particular, on the status of RACF resources. 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.

If you do not use program control to control access to DSMON, you must have Auditor authority to run DSMON. If you do use program control to control access to DSMON, you must have sufficient authority in the access list of program ICHDSM00. Also, you must have READ access authority to the SYS1.PARMLIB.

DSMON is an important tool for auditing system integrity and data security controls. We recommend that all DSMON reports be produced on a weekly basis and reviewed by the auditor and security administrator staff as well as by the technical support staff, in order to detect possible security exposures quickly.

DSMON produces the following reports:

- System report
- Group Tree report
- Program Properties Table report
- RACF Authorized Caller Table report
- RACF Class Description Table report
- RACF Exits report
- RACF Global Access Checking Table report
- RACF Started Procedures Table report
- Selected User Attribute report
- Selected User Attribute Summary report
- Selected Data Set report

Note: Producing the Group Tree, Selected User Attribute and Selected User Attribute Summary report can have an impact on system performance. Depending on the size and load on your RACF databases, you should consider running these DSMON reports during slack time.

You can use the JCL included in job RACDSM to run DSMON. The input to DSMON consists of the SYS1.PARMLIB data set. A SYSIN DD statement also lets you specify DSMON control statements for selected functional reports. With DSMON control statements, you can specify the reports that you want and the number of lines per page.

A.19, “DSMON” on page 164 contains an example of a job to run the RACF DSMON, and Appendix D, “DSMON Reports Output Samples” on page 175 contains sample output reports generated by DSMON.

8.2.1 Control Statements

DSMON has three control statements: LINECOUNT, FUNCTION, and USEROPT. Each is discussed below.

LINECOUNT Specifies the number of lines per page for reports. If you do not specify a linecount, the default value is 55 lines per page. The valid values 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.

FUNCTION Function-name Specifies the DSMON functional report that you want to produce. The default value is ALL, which causes DSMON to generate all reports except USRDSN. You can specify any of the following:

SYSTEM System report - This report contains the identification number and model of the CPU; the name, version, and release of the operating system; the serial number of the system residence volume; the system identifier that SMF uses; the RACF version and release number and whether RACF is active. If RACF is inactive, DSMON prints a message that tells you whether RACF was not activated at IPL or was deactivated by the RVARY command. You can use the System report to verify that the system has the expected hardware and software. The System report is the only one that DSMON produces when RACF is inactive.

RACGRP Group Tree report - This report lists all subgroups for the SYS1 group, as well as the owner or each group listed in the report. Alternately, if a USEROPT RACGRP control statement is used with the FUNCTION RACGRP statement, this report lists the subgroup hierarchy beginning with the group specified on the USEROPT RACGRP statement. You can use the Group Tree report to examine all overall RACF group structure for your system.

SYSPPT Program Properties Table (PPT) report - This report lists all the programs in the PPT and indicates whether each program is authorized to bypass password protection and whether it runs in a system key.

RACAUT RACF Authorized Caller Table report - This report lists the names of all programs in the RACF authorized caller table; that is, programs that are authorized to issue RACINIT (which performs user verification) or RACLIST (which load profiles into main storage) requests, or both.

RACCDT RACF Class Descriptor Table report - This report lists, for each general resource class, the class name, the default UACC, whether the class is active, whether auditing is being done, whether statistics are being kept, and whether Operations attribute users have access.

RACEXT RACF Exits report - This report lists the names of all RACF exits that are in active in the operating system and specifies the size of each module.

RACGAC RACF Global Access Checking Table report - This report lists all entries in the GAC table. Each entry consists of a resource name and its associated global access authority level.

RACSPT RACF Started Procedures Table report - This report lists each entry in the SPT: the procedure name, userid, and group name to be associated with the procedure; and whether the procedure is privileged or trusted. Trusted means the same as privileged, except that auditing can be requested by using the SETROPTS LOGOPTIONS command.

RACUSR RACF Selected User Attribute report - This report lists all RACF users with Special, Operations, Auditor or Revoke authority, and whether the user has the attribute at the system or group level.

SYSAPF This report lists the names of the APF authorized libraries.

SYSLNK This report lists the names of the libraries in LINKLST.

SYSCAT This report lists the names of the MVS master catalog and all user catalogs.

RACDST This report lists the primary and backup RACF database names.

SYSSDS This report lists selected system data sets.

USRDSN This option must be used with the USEROPT USRDSN control statement. It produces a report listing the data sets specified on the USEROPT USRDSN statement and their security attributes.

USEROPT Function-name: Specifies the DSMON user option report that you want to produce. In function-name, you can specify the following options:

RACGRP GROUP-NAME This option must be used with the FUNCTION RACGRP control statement. It produces a report listing the subgroup hierarchy beginning with the group specified by group-name.

USRDSN DSNAME Selected Users Data Set report - This option must be used with the FUNCTION USRDSN control statement. It produces a report listing the data set specified by dsname and its security attributes.

8.2.2 DSMON FUNCTION Statement Examples

Figure 26 contains FUNCTION statement. That will produce a DSMON report listing:

- All the entries in the GAC table
- The active RACF exit routines
- The names of all the APF libraries specified in IEAAPFxx in SYS1.PARMLIB

```
FUNCTION RACGAC
FUNCTION RACEXT
FUNCTION SYSAPF
```

Figure 26. FUNCTION Statement Example

8.2.3 DSMON USEROPT Statement Examples

Figure 27 contains the USEROPT statement. From the statements in the example, DSMON will produce both:

- A selected data set report for ′PAYROLL.TEST.OBJ′
- A Group Tree report starting with the group PAYROLL

```
FUNCTION USRDSN
 USEROPT USRDSN PAYROLL.TEST.OBJ

FUNCTION RACGRP
 USEROPT RACGRP PAYROLL
```

Figure 27. USEROPT Statement Examples

Section A.19, “DSMON” on page 164 contains example of a job to run RACF DSMON reports.

Chapter 9. RACF ISPF Panels

After the planning for RACF implementation has taken place, you can perform security and group administration tasks by using various RACF commands. For example, you can define new users using the ADDUSER command, define new groups using the ADDGROUP command, and so on. The RACF commands include various operands to perform the different tasks required.

As an alternative to using RACF commands to perform administration tasks, you can use the ISPF panels (assuming that ISPF is installed at your location). Using the panels, you need not memorize command or operand names, you need only to fill in the appropriate information on the proper panels.

The RACF commands provide the following advantages:

- Entering commands can be faster than displaying many panels in sequence.
- The examples in the manuals are generally command example. So, using commands from the manual descriptions should be relatively straightforward.
- Getting online help is easier. To see online help for the PERMIT command, for example, enter HELP PERMIT.
- Getting explanations for messages beginning with ICH is easier. For example, to see the explanation for message ICH06001I, enter HELP PERMIT MSGID(ICH06001I). If you get a message, but not a message id, enter PROFILE MSGID.

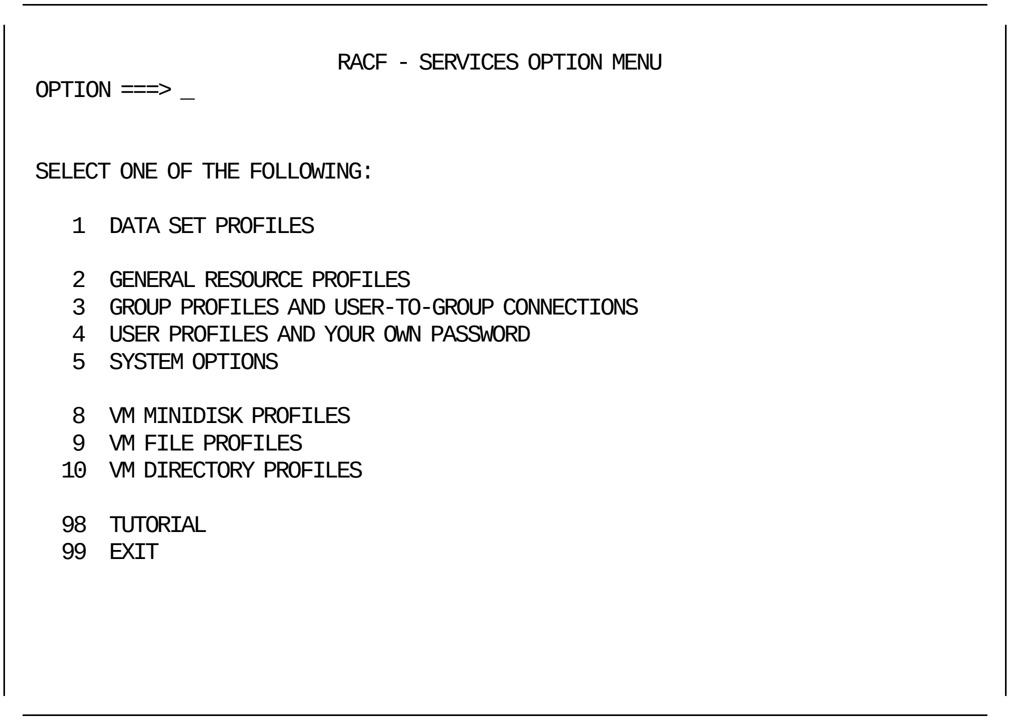
The ISPF panels provide the following advantages:

- ISPF creates a summary record in the ISPF log of the work that you do, unless you use the TSO Session Manager.
- From the panels you can press the HELP key to display a brief description of the fields on the panel.
- The option chosen when installing RACF panels will determine whether output from listings and search results is displayed in a scrollable form.
- The ISPF panels for working with password rules allow you to enter all the password rules on one panel.

To use the RACF from the ISPF panels, you must enter ISPF. The primary option menu appears. Choose option “R” for RACF.

This is the usual way to access RACF panels, but your installation may have implemented a different path. Check with your system programmer for more information.

You will see the RACF menu shown in Figure 28



```
                       RACF - SERVICES OPTION MENU
OPTION ===> _


SELECT ONE OF THE FOLLOWING:

   1  DATA SET PROFILES

   2  GENERAL RESOURCE PROFILES
   3  GROUP PROFILES AND USER-TO-GROUP CONNECTIONS
   4  USER PROFILES AND YOUR OWN PASSWORD
   5  SYSTEM OPTIONS

   8  VM MINIDISK PROFILES
   9  VM FILE PROFILES
  10  VM DIRECTORY PROFILES

  98  TUTORIAL
  99  EXIT
```

Figure 28. RACF Service Option Menu

From any panel, pressing PF1 displays a Help panel.

If you have a question about the information you should provide on the panel, type help on the command line.

If you need the panel ID for diagnosis purposes, enter the PANELID command and the panel identification will appear in the left part of the screen.

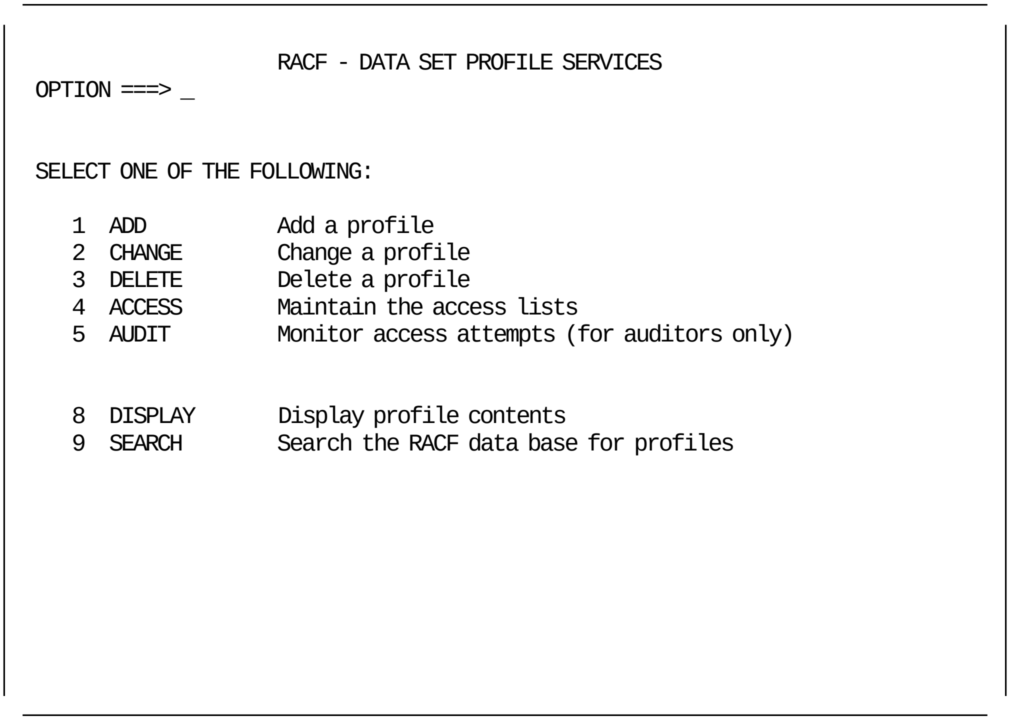
You can use the split screen option to view the tutorial and the panel at the same time. To do this, type split on the command line, then type swap on the command line when you want to change between the two screens. Choose the tutorial option on one screen and the RACF menu on the other screen.

9.1 Data Set Profiles

When you select option 1 from the RACF Services Options menu, the Data Set Profile Services menu is displayed on your screen (Figure 29).

Through this panel, you can select options to create, change, and delete your data set profiles. This screen can also be used to maintain the access list and to search the profiles.

The following sections describe how to add data set profiles and to maintain the access lists using ISPF panels. ISPF panels to display the contents of profiles and to search the RACF data base are included. Options 2 e 3 of the RACF Data Set Profile Services show panels similar to the panels shown by option 1, and will not be discussed in this manual, as well as option 5.



```
                    RACF - DATA SET PROFILE SERVICES
OPTION ===> _


SELECT ONE OF THE FOLLOWING:

   1  ADD          Add a profile
   2  CHANGE       Change a profile
   3  DELETE       Delete a profile
   4  ACCESS       Maintain the access lists
   5  AUDIT        Monitor access attempts (for auditors only)


   8  DISPLAY      Display profile contents
   9  SEARCH       Search the RACF data base for profiles
```

Figure 29. Data Set Profile Services Menu

9.1.1 Adding a Data Set Profile

The first option of the RACF Data Set Profile Services is used to create generic or discrete data set profiles.

If you are creating a generic profile, and its name contains no generic character (*, **, or %) you have to specify YES for the GENERIC field.

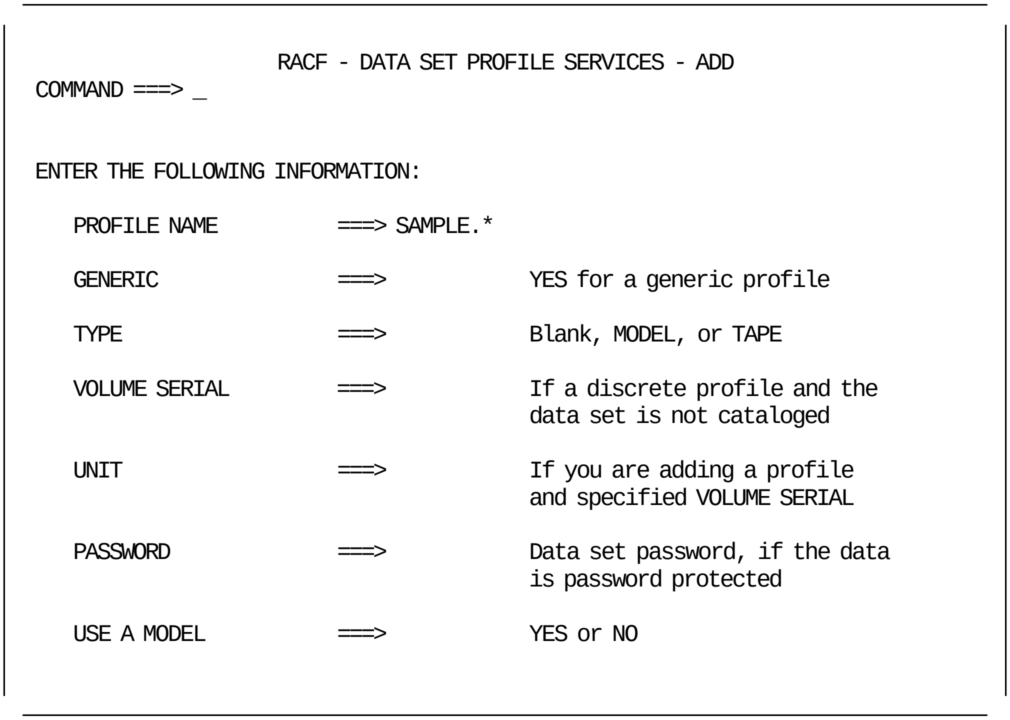
The VOLUME and TYPE fields are required for discrete profiles that protect uncataloged data sets.

You can also use another profile as a model when creating a profile. Specify YES in the USE A MODEL field.

Figure 30 shows an example of defining a generic profile SAMPLE.*.

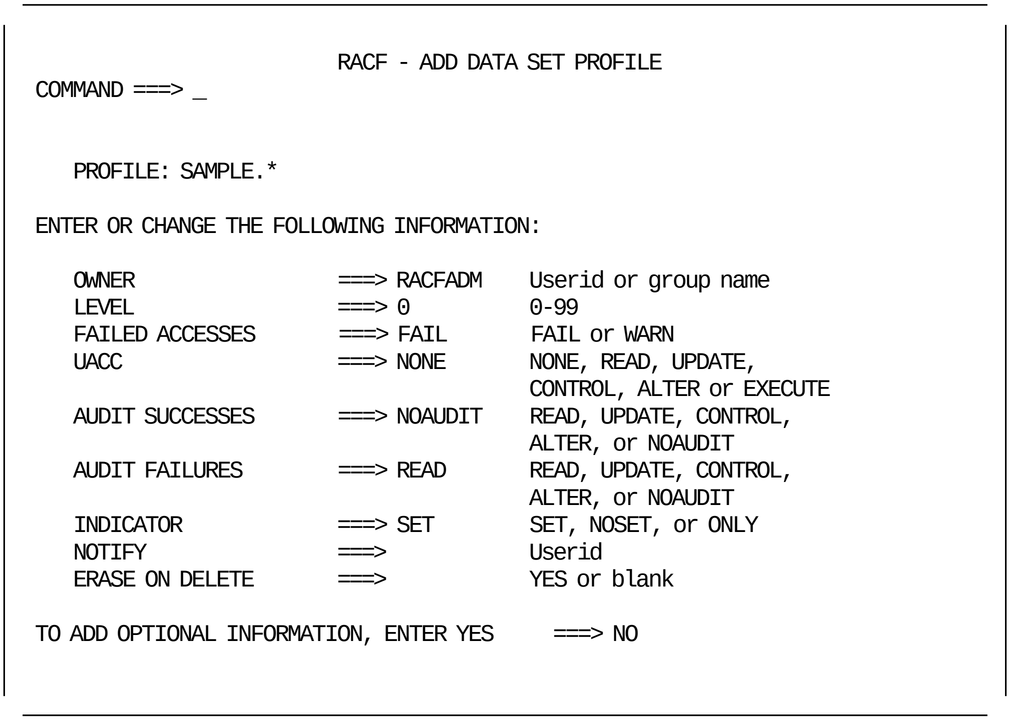
After you have entered the information on the panel in Figure 30, you will receive the panel shown in Figure 31. Use that panel to specify the profile contents, such as the owner, the UACC, the level of accesses you want to log, the user to be notified if an unauthorized attempt is made to access a resource protected by this profile, and other information.

For example, Figure 31 specifies, among other things, that the profile SAMPLE.* has a UACC of NONE, is owned by RACFADM, and will have all unauthorized accesses logged, but no user will be notified when an unauthorized access occur.



```
                    RACF - DATA SET PROFILE SERVICES - ADD
COMMAND ===> _


ENTER THE FOLLOWING INFORMATION:

   PROFILE NAME         ===> SAMPLE.*

   GENERIC              ===>           YES for a generic profile

   TYPE                 ===>           Blank, MODEL, or TAPE

   VOLUME SERIAL        ===>           If a discrete profile and the
                                       data set is not cataloged

   UNIT                 ===>           If you are adding a profile
                                       and specified VOLUME SERIAL

   PASSWORD             ===>           Data set password, if the data
                                       is password protected

   USE A MODEL          ===>           YES or NO
```

Figure 30. Adding a Data Set Profile - Part 1



```
                         RACF - ADD DATA SET PROFILE
COMMAND ===> _


   PROFILE: SAMPLE.*

ENTER OR CHANGE THE FOLLOWING INFORMATION:

   OWNER                ===> RACFADM   Userid or group name
   LEVEL                ===> 0         0-99
   FAILED ACCESSES      ===> FAIL      FAIL or WARN
   UACC                 ===> NONE      NONE, READ, UPDATE,
                                       CONTROL, ALTER or EXECUTE
   AUDIT SUCCESSES      ===> NOAUDIT   READ, UPDATE, CONTROL,
                                       ALTER, or NOAUDIT
   AUDIT FAILURES       ===> READ      READ, UPDATE, CONTROL,
                                       ALTER, or NOAUDIT
   INDICATOR            ===> SET       SET, NOSET, or ONLY
   NOTIFY               ===>           Userid
   ERASE ON DELETE      ===>           YES or blank

TO ADD OPTIONAL INFORMATION, ENTER YES     ===> NO
```

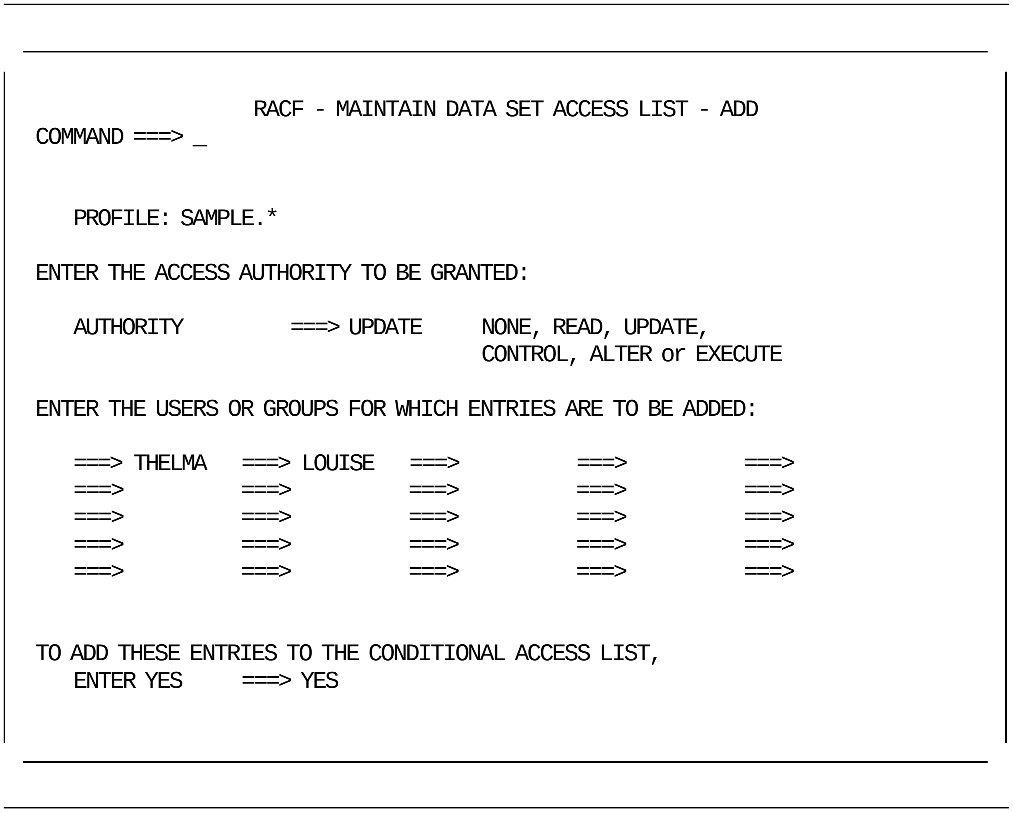
Figure 31. Adding a Data Set Profile - Panel 2

9.1.2 Updating the Access List

Use the ACCESS option in the Data Set Profile Services menu to add or delete users or groups from the access list of a data set profile.

The first panel that appears is similar to that shown in Figure 30. After you enter the profile name, you can either add or remove users or groups from the access list of the profile.

Figure 32 shows an example of including users THELMA and LOUISE with access authority of UPDATE to the profile SAMPLE.*.

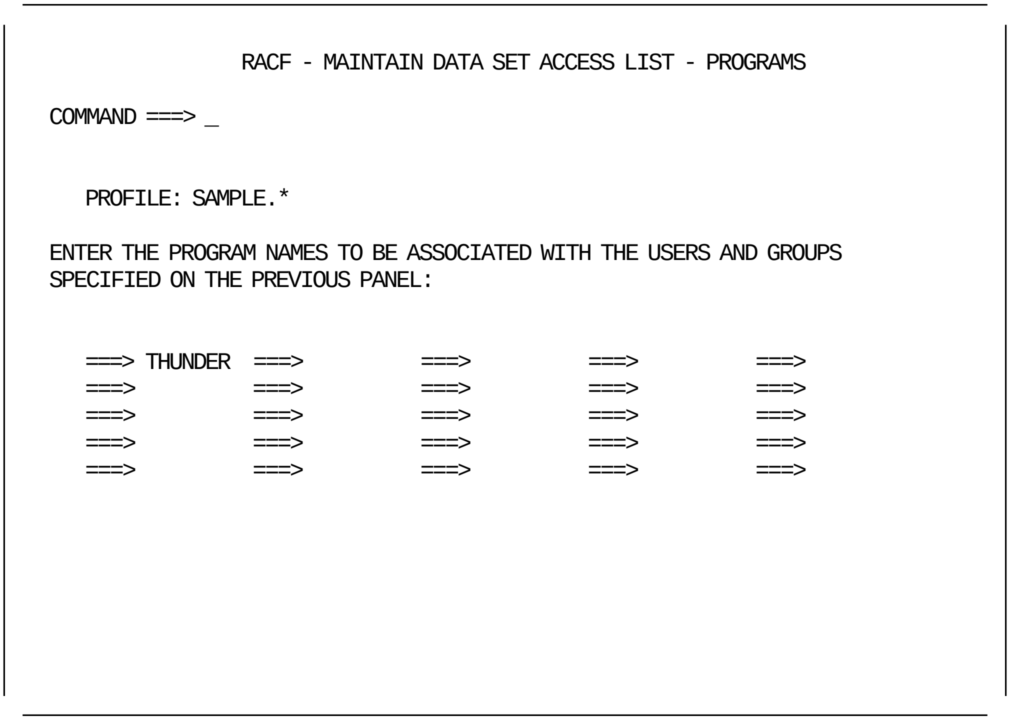


```
                    RACF - MAINTAIN DATA SET ACCESS LIST - ADD
COMMAND ===> _


   PROFILE: SAMPLE.*

ENTER THE ACCESS AUTHORITY TO BE GRANTED:

   AUTHORITY         ===> UPDATE     NONE, READ, UPDATE,
                                     CONTROL, ALTER or EXECUTE

ENTER THE USERS OR GROUPS FOR WHICH ENTRIES ARE TO BE ADDED:

   ===> THELMA   ===> LOUISE   ===>          ===>          ===>
   ===>          ===>          ===>          ===>          ===>
   ===>          ===>          ===>          ===>          ===>
   ===>          ===>          ===>          ===>          ===>
   ===>          ===>          ===>          ===>          ===>


TO ADD THESE ENTRIES TO THE CONDITIONAL ACCESS LIST,
   ENTER YES     ===> YES
```

Figure 32. Adding Users to the Access List

You can also create a conditional access list by specifying the name of the programs that these users must be running to have the specified access. Note that in Figure 32, when defining the users, a YES was specified in the last field, which means that a conditional access list is to be created.

Figure 33 specifies, for example, that users THELMA and LOUISE will be able to access the data sets protected by the profile SAMPLE.* only when running program THUNDER.



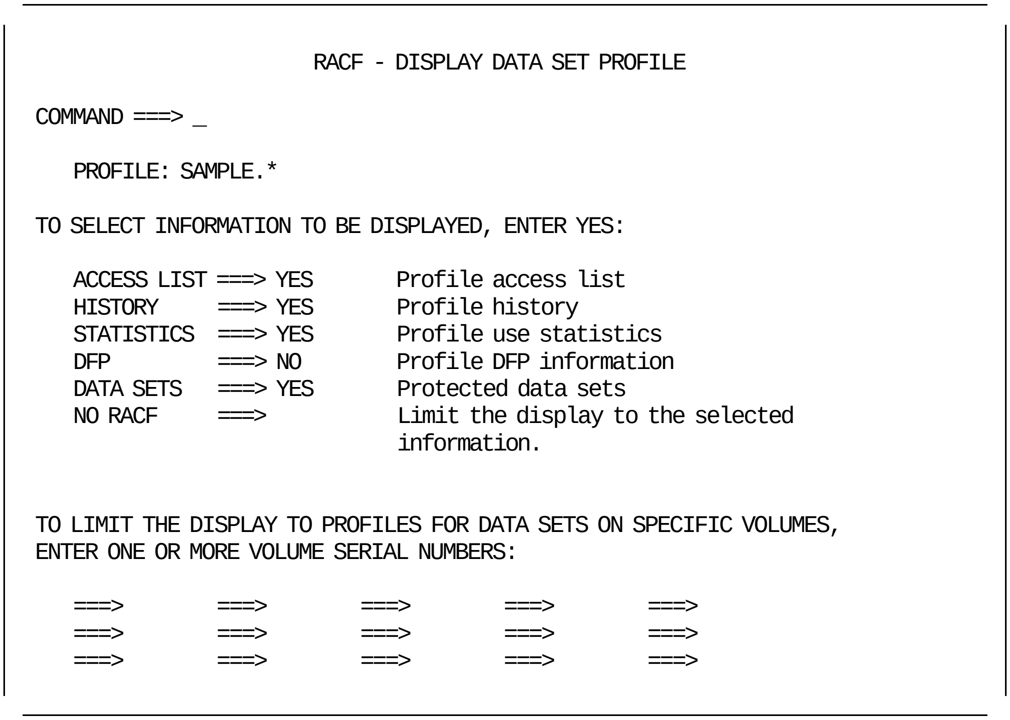
```
                 RACF - MAINTAIN DATA SET ACCESS LIST - PROGRAMS

COMMAND ===> _


   PROFILE: SAMPLE.*

ENTER THE PROGRAM NAMES TO BE ASSOCIATED WITH THE USERS AND GROUPS
SPECIFIED ON THE PREVIOUS PANEL:


   ===> THUNDER  ===>          ===>          ===>          ===>
   ===>          ===>          ===>          ===>          ===>
   ===>          ===>          ===>          ===>          ===>
   ===>          ===>          ===>          ===>          ===>
   ===>          ===>          ===>          ===>          ===>
```

Figure 33. Creating a Conditional Access List

9.1.3 Displaying Profile Contents

Select the DISPLAY option in the RACF Data Set Profile Services panel (Figure 29) to display information about a data set profile. You can select the options you want to be displayed by using the panel shown in Figure 34.

In this example, all information about the profile is requested, except that related to DFP. See Figure 35 to see the result of this request.



```
                          RACF - DISPLAY DATA SET PROFILE

COMMAND ===> _

   PROFILE: SAMPLE.*

TO SELECT INFORMATION TO BE DISPLAYED, ENTER YES:

   ACCESS LIST ===> YES       Profile access list
   HISTORY     ===> YES       Profile history
   STATISTICS  ===> YES       Profile use statistics
   DFP         ===> NO        Profile DFP information
   DATA SETS   ===> YES       Protected data sets
   NO RACF     ===>           Limit the display to the selected
                              information.


TO LIMIT THE DISPLAY TO PROFILES FOR DATA SETS ON SPECIFIC VOLUMES,
ENTER ONE OR MORE VOLUME SERIAL NUMBERS:

   ===>        ===>        ===>        ===>        ===>
   ===>        ===>        ===>        ===>        ===>
   ===>        ===>        ===>        ===>        ===>
```

Figure 34. Selecting Display Options

Figure 34 shows the result of the display requested using the Display Data Set Profile panel. Note, among other information, the UACC, the owner of the profile, the auditing option of listing unauthorized attempts to access the data sets, and the conditional access list.

```
INFORMATION FOR DATASET SAMPLE.* (G)

LEVEL  OWNER    UNIVERSAL ACCESS   WARNING   ERASE
-----  -------- ----------------   -------   -----
 00    RACFADM         NONE          NO      NO

AUDITING
--------
FAILURES(READ)

NOTIFY
--------
NO USER TO BE NOTIFIED

YOUR ACCESS  CREATION GROUP  DATASET TYPE
-----------  --------------  ------------
   ALTER        TSO            NON-VSAM

NO INSTALLATION DATA

              SECURITY LEVEL
------------------------------------------
NO SECURITY LEVEL

CATEGORIES
----------
NO CATEGORIES

SECLABEL
--------
NO SECLABEL

CREATION DATE  LAST REFERENCE DATE  LAST CHANGE DATE
(DAY) (YEAR)        (DAY) (YEAR)      (DAY) (YEAR)
-------------  -------------------  ----------------
 100    92      NOT APPLICABLE FOR GENERIC PROFILE

ALTER COUNT  CONTROL COUNT  UPDATE COUNT  READ COUNT
-----------  -------------  ------------  ----------
NOT APPLICABLE FOR GENERIC PROFILE

   ID     ACCESS
--------  -------
NO ENTRIES IN STANDARD ACCESS LIST

   ID    ACCESS   CLASS          ENTITY NAME
-------- ------- -------- --------------------------
LOUISE   UPDATE  PROGRAM  THUNDER
THELMA   UPDATE  PROGRAM  THUNDER

DATA SETS AFFECTED BY PROFILE CHANGE
------------------------------------
SAMPLE.DATA
SAMPLE.JCL
```

Figure 35. Data Set Profile Contents

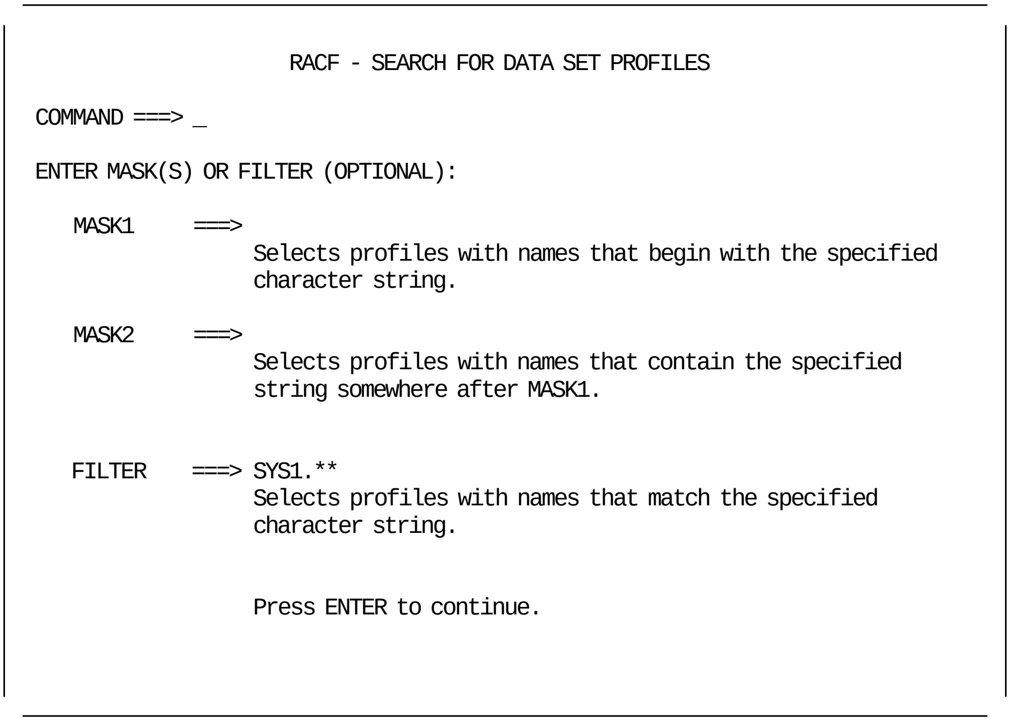
9.1.4 Searching for Data Set Profiles

Using option 9 in the RACF Data Set Profile Services panel (Figure 29 on page 93) you can search data set profiles defined to RACF.

You can use either the MASK1 and MASK2 operands, or the SEARCH operand. You cannot specify both MASK and FILTER. The FILTER operand allows you to use the generic characters (*, **, and %) in the character string of your search.

If you do not specify any entries in the fields, the system assumes * in the mask and lists all the profiles that your user ID can list. If you have Special authority, all data set profiles will be listed.

The example shown in Figure 36 displays a list of all profiles your user ID can list, whose high-level qualifier is SYS1.

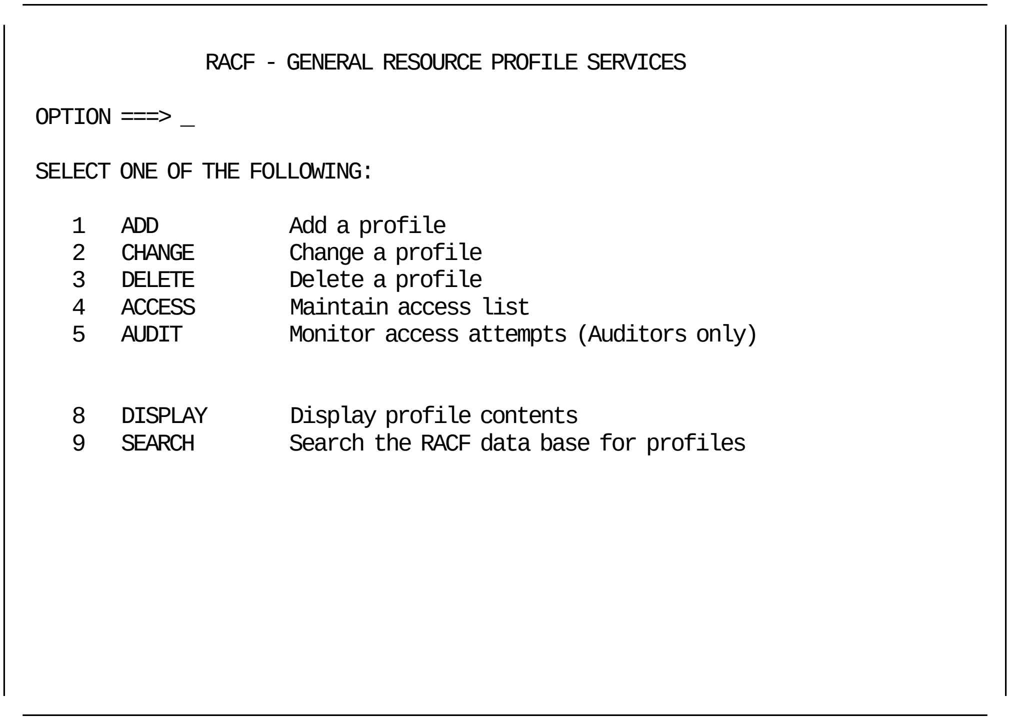


```
                       RACF - SEARCH FOR DATA SET PROFILES

COMMAND ===> _

ENTER MASK(S) OR FILTER (OPTIONAL):

   MASK1     ===>
                  Selects profiles with names that begin with the specified
                  character string.

   MASK2     ===>
                  Selects profiles with names that contain the specified
                  string somewhere after MASK1.


   FILTER    ===> SYS1.**
                  Selects profiles with names that match the specified
                  character string.


                  Press ENTER to continue.
```

Figure 36. Searching for Data Set Profiles

9.2 General Resource Profiles

ISPF panels also can be used to maintain activity in general resource profiles. Select option 2 from the RACF Service Options menu, and the panel shown in Figure 37 will be displayed on your screen. This panel can be used to call other panels to create, change, delete, and look for general resource profiles, as well as to perform maintenance in the access lists.



```
               RACF - GENERAL RESOURCE PROFILE SERVICES

OPTION ===> _

SELECT ONE OF THE FOLLOWING:

   1   ADD          Add a profile
   2   CHANGE       Change a profile
   3   DELETE       Delete a profile
   4   ACCESS       Maintain access list
   5   AUDIT        Monitor access attempts (Auditors only)


   8   DISPLAY      Display profile contents
   9   SEARCH       Search the RACF data base for profiles
```

Figure 37. General Resource Profile Services Menu

The General Resource Services panels can be used for any general resource class, such as TERMINAL, DASDVOL, and others. The following discussion uses the TERMINAL class to exemplify how to add a general resource profile and display its contents. Options 2, 3, 5, and 9 are not covered in this manual.

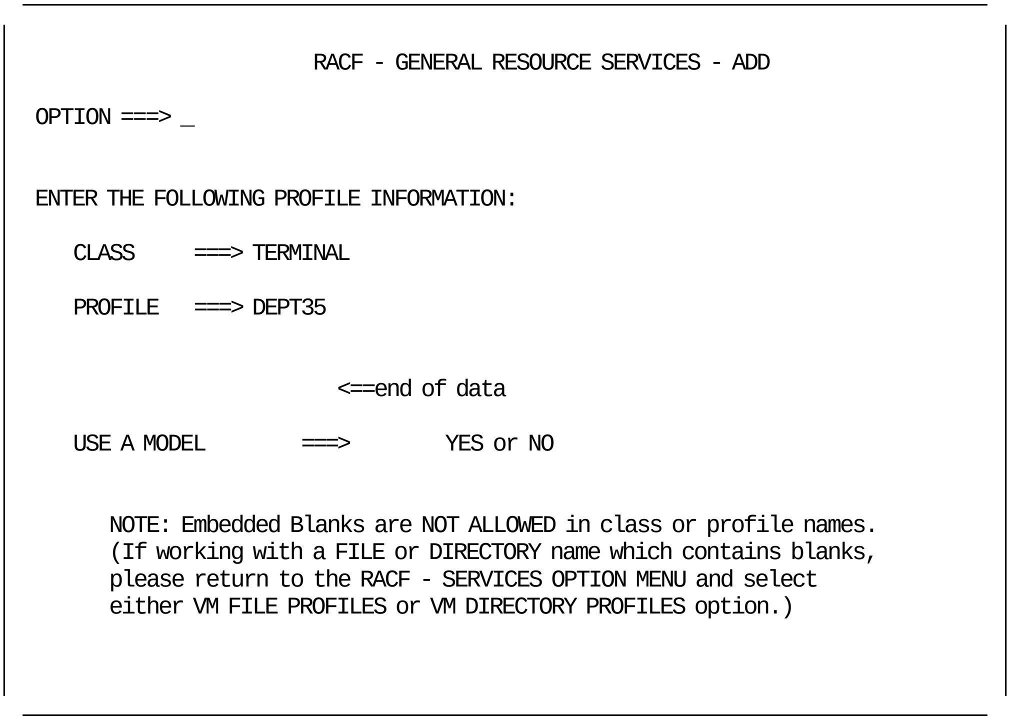
9.2.1 Adding a General Resource Profile

The first option in the RACF General Resource Profile Services creates general resource profiles. When defining this kind of profile, you will be asked for the name of the profile and the general resource class in which this profile will be created.

Figure 38 shows the ISPF panel used to name a profile and specify its class. In this example, a profile called DEPT35 is being created in the TERMINAL class.

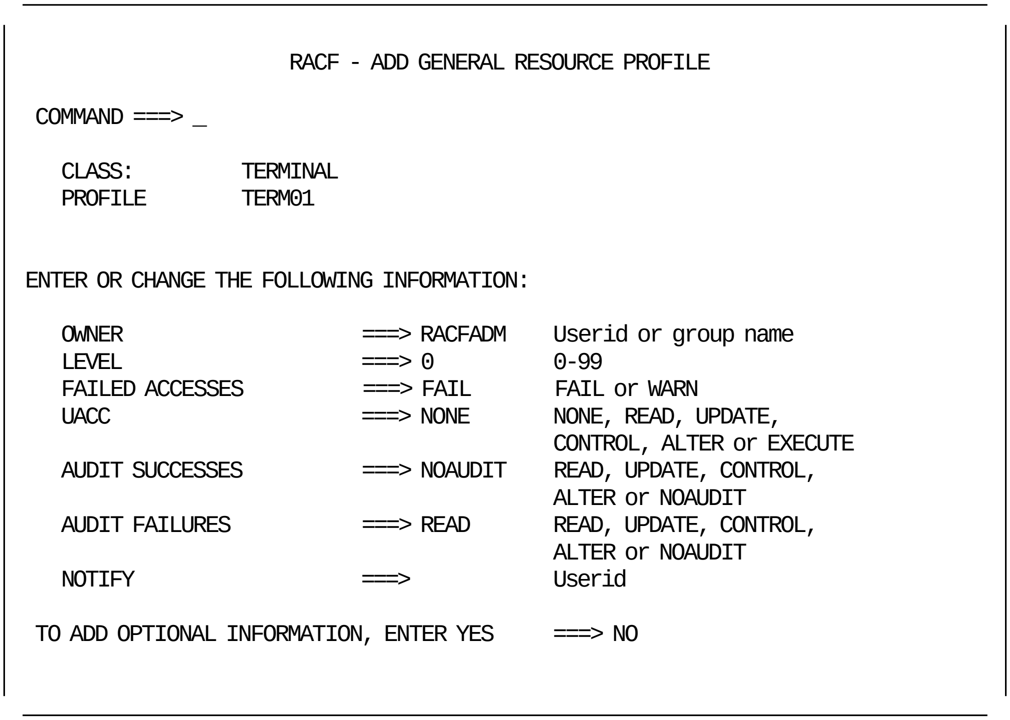
After that, you will receive the Add General Resource Profile panel, where you specify some information about the profile you are creating.

Figure 39 shows the sequence for defining the profile DEPT35. Note that the options are equivalent to those specified when defining data set profiles.



```
                    RACF - GENERAL RESOURCE SERVICES - ADD

OPTION ===> _


ENTER THE FOLLOWING PROFILE INFORMATION:

   CLASS     ===> TERMINAL

   PROFILE   ===> DEPT35


                         <==end of data

   USE A MODEL        ===>        YES or NO


      NOTE: Embedded Blanks are NOT ALLOWED in class or profile names.
      (If working with a FILE or DIRECTORY name which contains blanks,
      please return to the RACF - SERVICES OPTION MENU and select
      either VM FILE PROFILES or VM DIRECTORY PROFILES option.)
```

Figure 38. Adding a General Resource Profile - Panel 1



```
                    RACF - ADD GENERAL RESOURCE PROFILE

COMMAND ===> _

  CLASS:         TERMINAL
  PROFILE        TERM01


ENTER OR CHANGE THE FOLLOWING INFORMATION:

  OWNER                   ===> RACFADM    Userid or group name
  LEVEL                   ===> 0          0-99
  FAILED ACCESSES         ===> FAIL       FAIL or WARN
  UACC                    ===> NONE       NONE, READ, UPDATE,
                                          CONTROL, ALTER or EXECUTE
  AUDIT SUCCESSES         ===> NOAUDIT    READ, UPDATE, CONTROL,
                                          ALTER or NOAUDIT
  AUDIT FAILURES          ===> READ       READ, UPDATE, CONTROL,
                                          ALTER or NOAUDIT
  NOTIFY                  ===>            Userid

TO ADD OPTIONAL INFORMATION, ENTER YES    ===> NO
```

Figure 39. Adding a General Resource Profile - Panel 2

9.2.2 Updating the Access List

To add or delete users or groups from the access list of a general resource profile, use the ACCESS option in the General Resource Profile Services menu (Figure 37 on page 100).

Access lists in general resources profiles are maintained in the same way as data set profiles. The only difference is that you have to specify the name of the general resource class.

You can also create a conditional access list to specify, for example, that a user must use specific consoles in order to access the general resource protected by the profile.

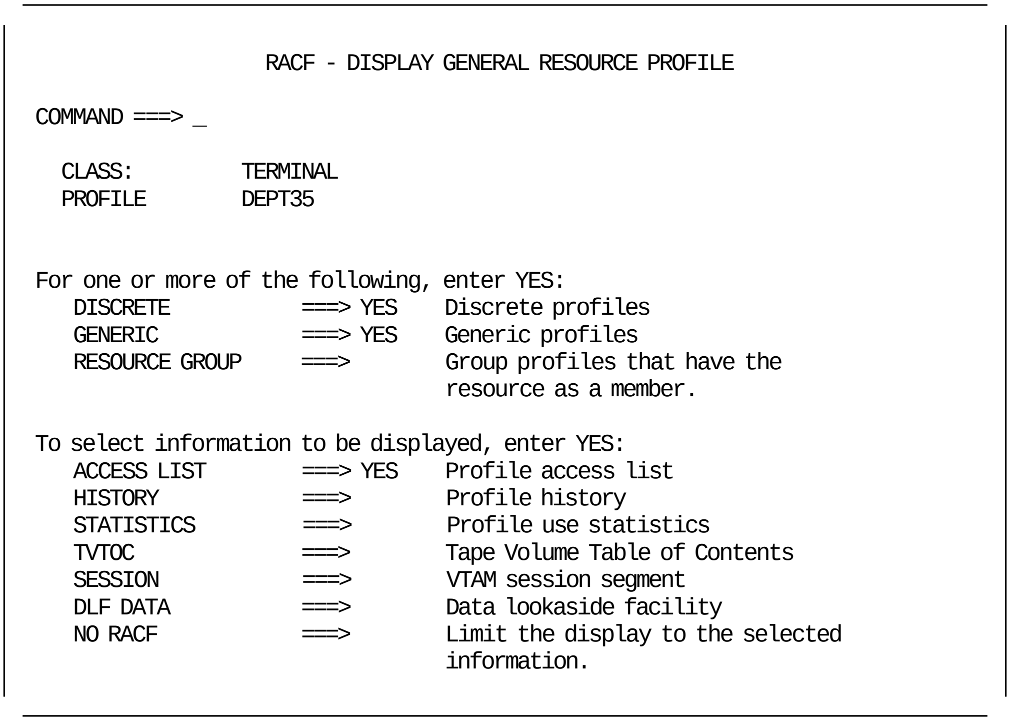
Refer to Section 9.1.2, “Updating the Access List” on page 95 to see how to perform maintenance on an access list.

9.2.3 Displaying Profile Contents

Select the DISPLAY option on the RACF General Resource Profile Services panel (Figure 37 on page 100) to obtain information about a general resource profile.

Figure 40 shows the information provided for profile DEPT35 in the TERMINAL class.

Figure 41 shows the result of this request. Note that, in this case, there is no conditional access list and this terminal can be used any day at any time.



```
                   RACF - DISPLAY GENERAL RESOURCE PROFILE

COMMAND ===> _

  CLASS:         TERMINAL
  PROFILE        DEPT35


For one or more of the following, enter YES:
   DISCRETE            ===> YES    Discrete profiles
   GENERIC             ===> YES    Generic profiles
   RESOURCE GROUP      ===>        Group profiles that have the
                                   resource as a member.

To select information to be displayed, enter YES:
   ACCESS LIST         ===> YES    Profile access list
   HISTORY             ===>        Profile history
   STATISTICS          ===>        Profile use statistics
   TVTOC               ===>        Tape Volume Table of Contents
   SESSION             ===>        VTAM session segment
   DLF DATA            ===>        Data lookaside facility
   NO RACF             ===>        Limit the display to the selected
                                   information.
```

Figure 40. Selecting Display Options

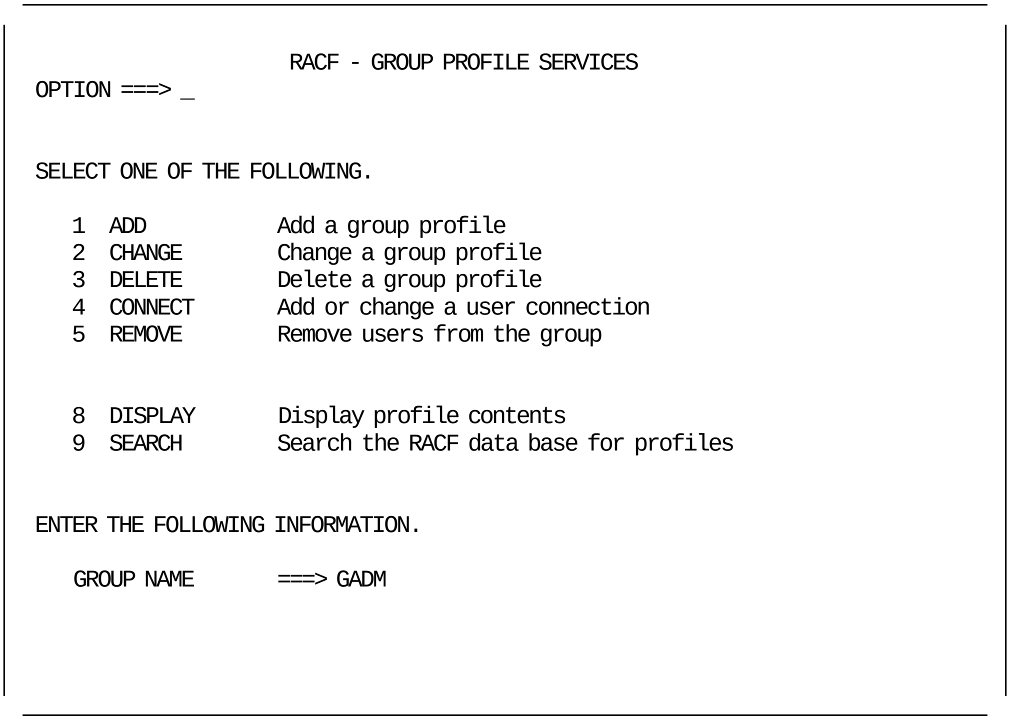
```
CLASS      NAME
-----      ----
TERMINAL   DEPT35

GROUP CLASS NAME
----- ----- ----
GTERMINL

LEVEL  OWNER      UNIVERSAL ACCESS  YOUR ACCESS  WARNING
-----  --------   ----------------  -----------  -------
 00    RACFADM         ALTER             ALTER    NO

INSTALLATION DATA
-----------------
NONE

APPLICATION DATA
----------------
NONE

SECLEVEL
--------
NO SECLEVEL

CATEGORIES
----------
NO CATEGORIES

SECLABEL
--------
NO SECLABEL

AUDITING
--------
FAILURES(READ)

TIMEZONE  LOGON ALLOWED   (DAYS)           (TIME)
--------  ---------------------------------------------
CPU TIME  ANYDAY                           ANYTIME

NOTIFY
------
NO USER TO BE NOTIFIED

USER      ACCESS   ACCESS COUNT
----      ------   ------ -----
GSYS      ALTER       000000
USR1      ALTER       000000

   ID     ACCESS  ACCESS COUNT  CLASS   ENTITY  NAME
-------- -------  ------ ----- -------- ------------
NO ENTRIES IN CONDITIONAL ACCESS LIST
```

Figure 41. General Resource Profile Contents

9.3 Group Profiles and User-to-group Connections

When you select option 3 from the RACF primary options menu, the Group Profile Services panel (Figure 42) is displayed. Through this panel you can create new groups, modify or delete existing groups, and connect or remove users from groups. This panel can also be used to list group information.

You must enter the name of the group to use any of the options except option 9 (Search group). In this discussion, we are using the GADM group as an example. Options 2, 3, and 5 are not discussed in this manual, since their panels are similiar to the panels shown by the other options.



```
                        RACF - GROUP PROFILE SERVICES
OPTION ===> _


SELECT ONE OF THE FOLLOWING.

   1  ADD           Add a group profile
   2  CHANGE        Change a group profile
   3  DELETE        Delete a group profile
   4  CONNECT       Add or change a user connection
   5  REMOVE        Remove users from the group


   8  DISPLAY       Display profile contents
   9  SEARCH        Search the RACF data base for profiles


ENTER THE FOLLOWING INFORMATION.

   GROUP NAME       ===> GADM
```

Figure 42. Group Profile Services

9.3.1 Adding New Groups

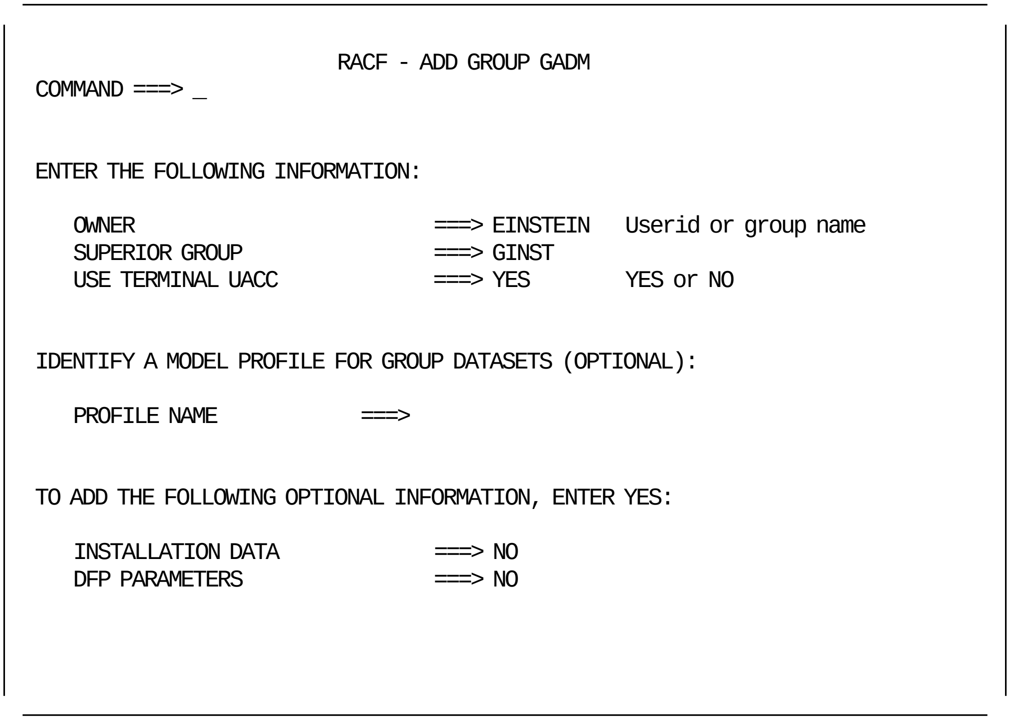
The first option of the RACF Group Profile Service is used to create new RACF groups. When you select this option, and fill in the GROUP NAME field, the panel shown in -- Fig ′FGRP1′ unknown -- is displayed.

On this panel, if you specify a group name in the owner field, you must specify the same group name in the SUPERIOR GROUP field. In the example, EINSTEIN is a user, so the superior group can have another name.

You can also use another group as a model when creating a new group. To do this, specify YES in the PROFILE NAME field.

Figure 43 shows an example of creating the group GADM. The name of the group was specified in the previous panel (Figure 42).

The RACF command used in this panel is ADDGROUP GADM SUPGROUP(GINST) OWNER(EINSTEIN).



```
                          RACF - ADD GROUP GADM
COMMAND ===> _


ENTER THE FOLLOWING INFORMATION:

   OWNER                          ===> EINSTEIN   Userid or group name
   SUPERIOR GROUP                 ===> GINST
   USE TERMINAL UACC              ===> YES        YES or NO


IDENTIFY A MODEL PROFILE FOR GROUP DATASETS (OPTIONAL):

   PROFILE NAME             ===>


TO ADD THE FOLLOWING OPTIONAL INFORMATION, ENTER YES:

   INSTALLATION DATA              ===> NO
   DFP PARAMETERS                 ===> NO
```

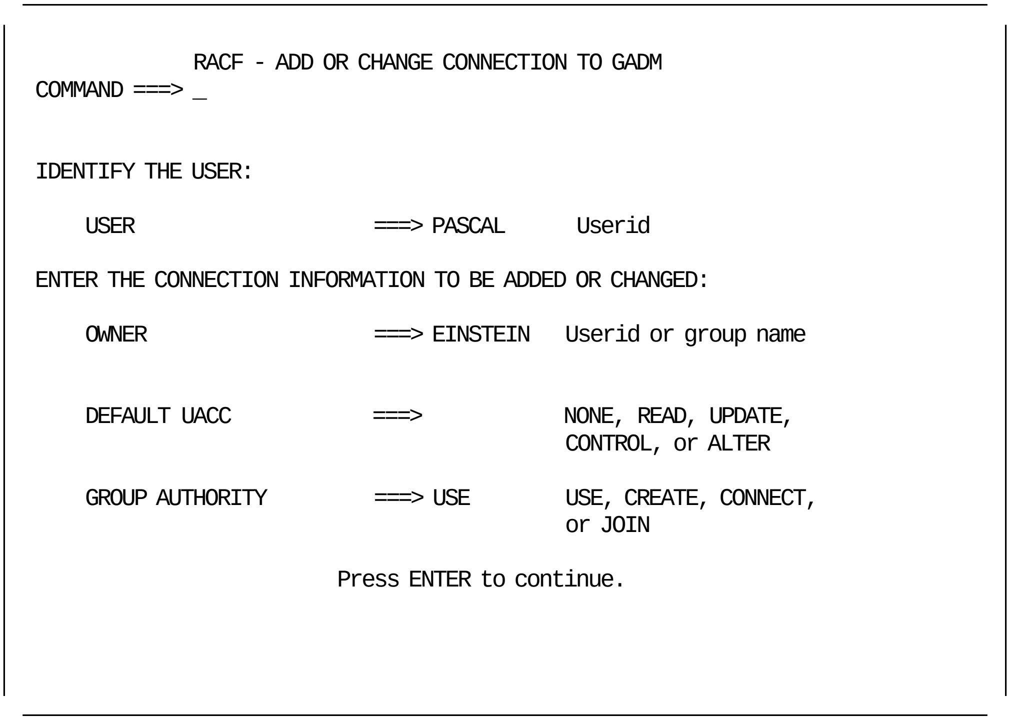
Figure 43. Adding a New Group

9.3.2 Connecting Users to Groups

The fourth option of the RACF Group Profile Services is used to connect users to groups and to change users connection to groups.

Figure 44 is an example of connecting user PASCAL to the GADM group with use authority. The name of the group was specified in the previous panel (Figure 42).

User ID EINSTEIN is the owner of the connect profile being defined. The owner was the user who is now using the panel (this value is the default) but this information is optional. The RACF command used in the panel is CONNECT PASCAL GROUP(GADM) OWNER(EINSTEIN).



```
              RACF - ADD OR CHANGE CONNECTION TO GADM
COMMAND ===> _


IDENTIFY THE USER:

    USER                   ===> PASCAL      Userid

ENTER THE CONNECTION INFORMATION TO BE ADDED OR CHANGED:

    OWNER                  ===> EINSTEIN   Userid or group name


    DEFAULT UACC           ===>            NONE, READ, UPDATE,
                                           CONTROL, or ALTER

    GROUP AUTHORITY        ===> USE        USE, CREATE, CONNECT,
                                           or JOIN

                          Press ENTER to continue.
```

Figure 44. Connecting User to Group

Another panel will request additional information. This data is optional; to connect a user to a group without privileges and special characteristics, just press ENTER on the panel.

9.3.3 Displaying Group Profile

Option 8 of the RACF Group Profile Services panel displays the RACF and DFP information about the group, like RACF LG command (list group).

Figure 45 is the output of listing RACF group GADM.

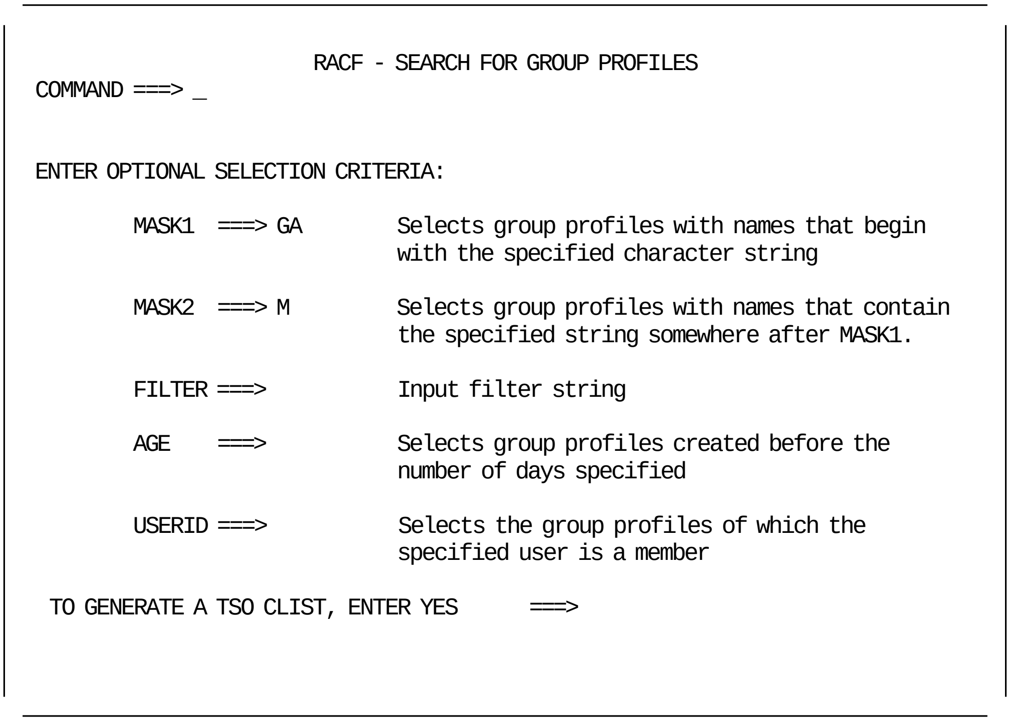
```
INFORMATION FOR GROUP GADM
    SUPERIOR GROUP=GINST        OWNER=EINSTEIN
    NO INSTALLATION DATA
    NO MODEL DATA SET
    TERMUACC
    NO SUBGROUPS
    USER(S)=      ACCESS=      ACCESS COUNT=      UNIVERSAL ACCESS=
      PASCAL        USE           000000              NONE
         CONNECT ATTRIBUTES=NONE
         REVOKE DATE=NONE                RESUME DATE=NONE
```

Figure 45. Listing for Group GADM

9.3.4 Searching for Group Profile

Option 9 allows you to search group profiles defined to RACF. In the example (Figure 46), the system lists the group profiles than have GA and M in the name. The search will list the GADM, GAME, and GATEM groups.

If you do not specify any entries in the fields, the system assumes * in the mask and lists all the profiles that your user ID can list. If you have Special authority, all the groups are listed.



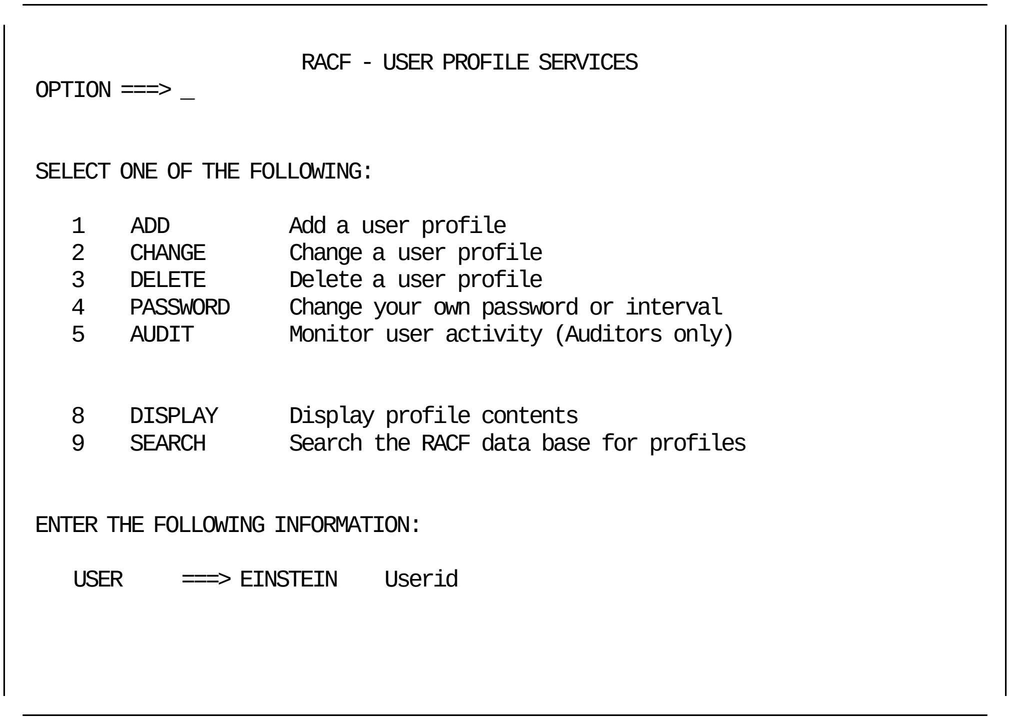
```
                        RACF - SEARCH FOR GROUP PROFILES
COMMAND ===> _


ENTER OPTIONAL SELECTION CRITERIA:

       MASK1  ===> GA        Selects group profiles with names that begin
                             with the specified character string

       MASK2  ===> M         Selects group profiles with names that contain
                             the specified string somewhere after MASK1.

       FILTER ===>           Input filter string

       AGE    ===>           Selects group profiles created before the
                             number of days specified

       USERID ===>           Selects the group profiles of which the
                             specified user is a member

 TO GENERATE A TSO CLIST, ENTER YES      ===>
```

Figure 46. Searching for Group Profile

9.4 User Profile and Your Own Password

When you select the option 4 of the RACF primary options menu (Figure 28 on page 92), the User Profile Service panel (Figure 47 on page 108) is displayed. Through this panel you can add new users, modify or delete existing users, change your password or monitor user activity. You need Special authority to use options 1 through 3, and you need Auditor authority to use option 5.

You must enter the name of the user to use any of the options except 4 (Password) and 9 (Search). In this discussion user ID EINSTEIN is used as an example for options 1 and 4.



```
                    RACF - USER PROFILE SERVICES
OPTION ===> _


SELECT ONE OF THE FOLLOWING:

   1   ADD          Add a user profile
   2   CHANGE       Change a user profile
   3   DELETE       Delete a user profile
   4   PASSWORD     Change your own password or interval
   5   AUDIT        Monitor user activity (Auditors only)


   8   DISPLAY      Display profile contents
   9   SEARCH       Search the RACF data base for profiles


ENTER THE FOLLOWING INFORMATION:

   USER     ===> EINSTEIN    Userid
```

Figure 47. User Profile Services Panel

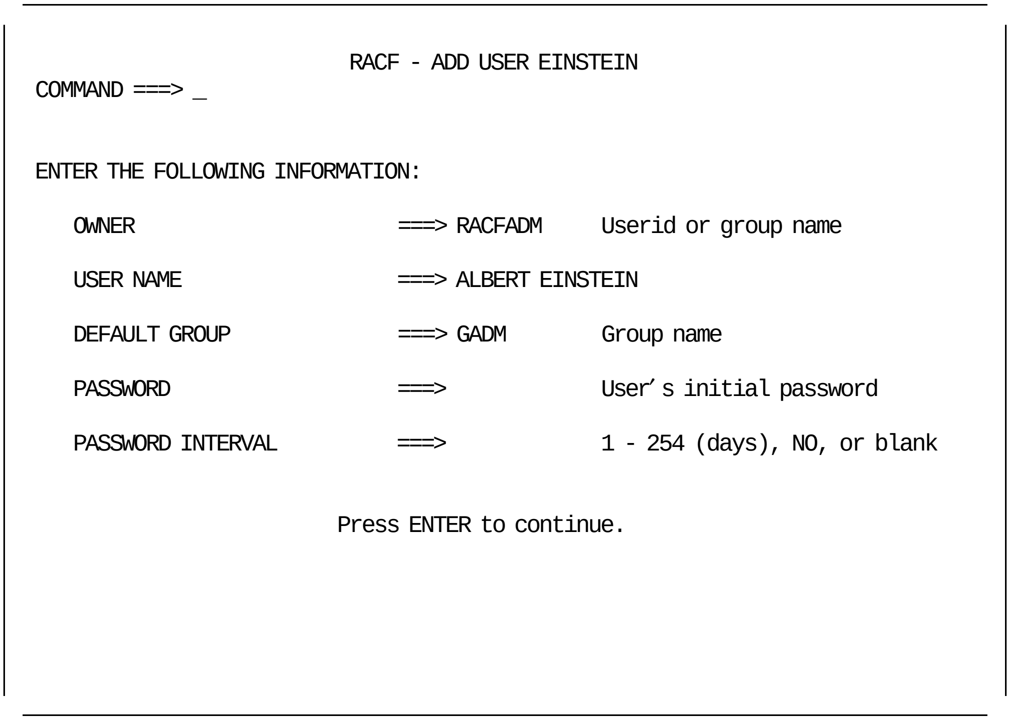
9.4.1 Adding New Users

The first option of the RACF User Profile Services panel creates new RACF users. Figure 48 shows an example of creating the user EINSTEIN. The name of the user was specified in the previous panel (Figure 47).

The owner of the profile is the user RACFADM; that is, the RACF administrator who is now creating the user. The default group of user EINSTEIN is GADM, the current connect group of the user unless the user requests another group when entering the system.

The initial password does not appear when you enter it on the panel. If you do not specify a password for the new user, the initial password is the name of the default group.

If you do not enter the password interval, the user′s password interval is the default defined in the SETROPTS options. If you specify an interval, the value must be less than the default of the SETROPTS option. If you specify a greater value, it will not take effect (the system uses the value specified in the SETROPTS option).



```
                         RACF - ADD USER EINSTEIN
COMMAND ===> _


ENTER THE FOLLOWING INFORMATION:

   OWNER                    ===> RACFADM     Userid or group name

   USER NAME                ===> ALBERT EINSTEIN

   DEFAULT GROUP            ===> GADM        Group name

   PASSWORD                 ===>             User′s initial password

   PASSWORD INTERVAL        ===>             1 - 254 (days), NO, or blank


                        Press ENTER to continue.
```

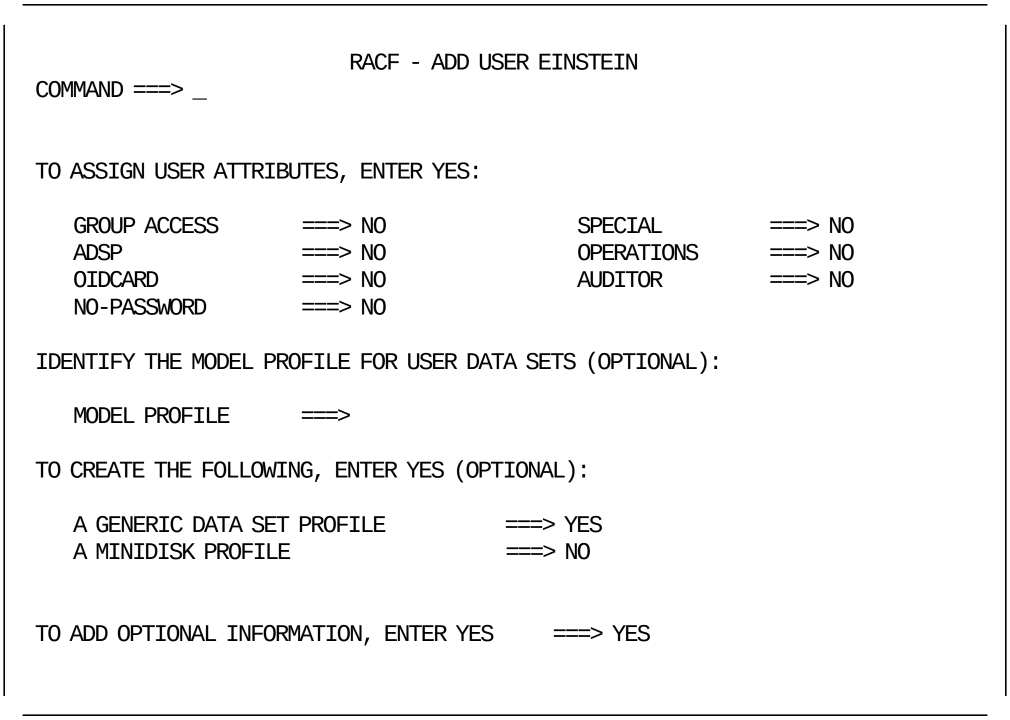
Figure 48. Adding New Users - Panel 1

You then have to specify the user ID characteristics.

In Figure 49, we specify YES in the GENERIC DATA SET PROFILE field so that the system creates a generic data set profile for the user. This generic data set profile, named EINSTEIN.*, protects all data sets where the high-level qualifier is the user′s user ID.

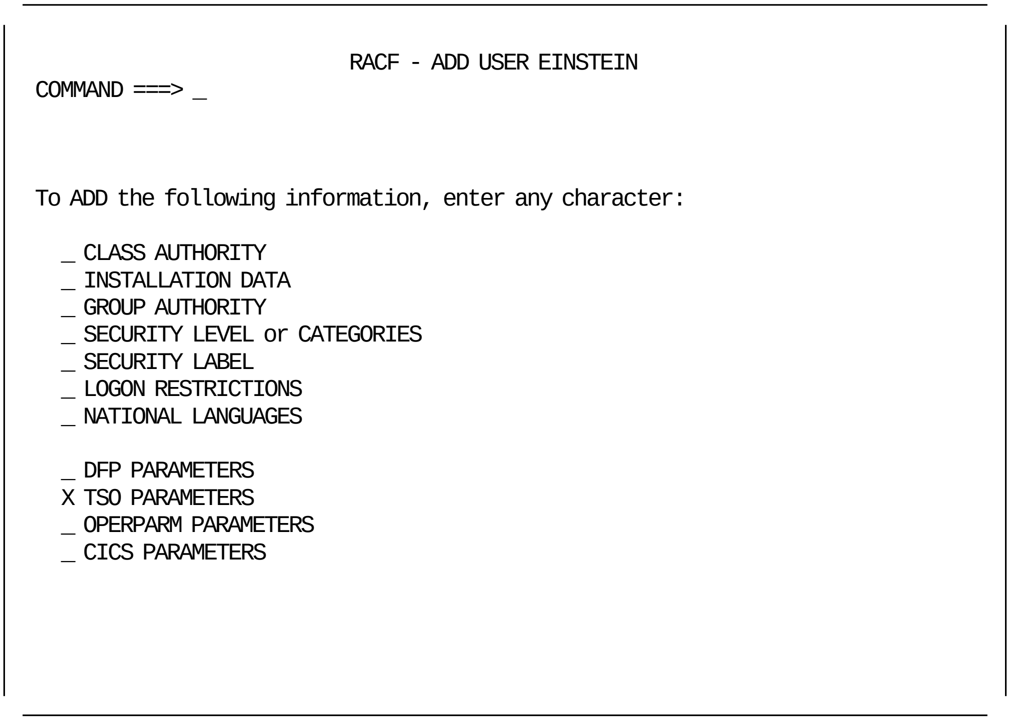
User ID EINSTEIN is going to use TSO, so we have to specify TSO parameters. To do that, enter YES in the ADD OPTIONAL INFORMATION field.

When you enter YES in the ADD OPTIONAL INFORMATION field, you have to choose which information you want to specify for the user. Figure 50 shows the options you have. We are only going to define EINSTEIN as a TSO user.



```
                          RACF - ADD USER EINSTEIN
COMMAND ===> _


TO ASSIGN USER ATTRIBUTES, ENTER YES:

   GROUP ACCESS      ===> NO               SPECIAL        ===> NO
   ADSP              ===> NO               OPERATIONS     ===> NO
   OIDCARD           ===> NO               AUDITOR        ===> NO
   NO-PASSWORD       ===> NO

IDENTIFY THE MODEL PROFILE FOR USER DATA SETS (OPTIONAL):

   MODEL PROFILE     ===>

TO CREATE THE FOLLOWING, ENTER YES (OPTIONAL):

   A GENERIC DATA SET PROFILE          ===> YES
   A MINIDISK PROFILE                  ===> NO


TO ADD OPTIONAL INFORMATION, ENTER YES     ===> YES
```

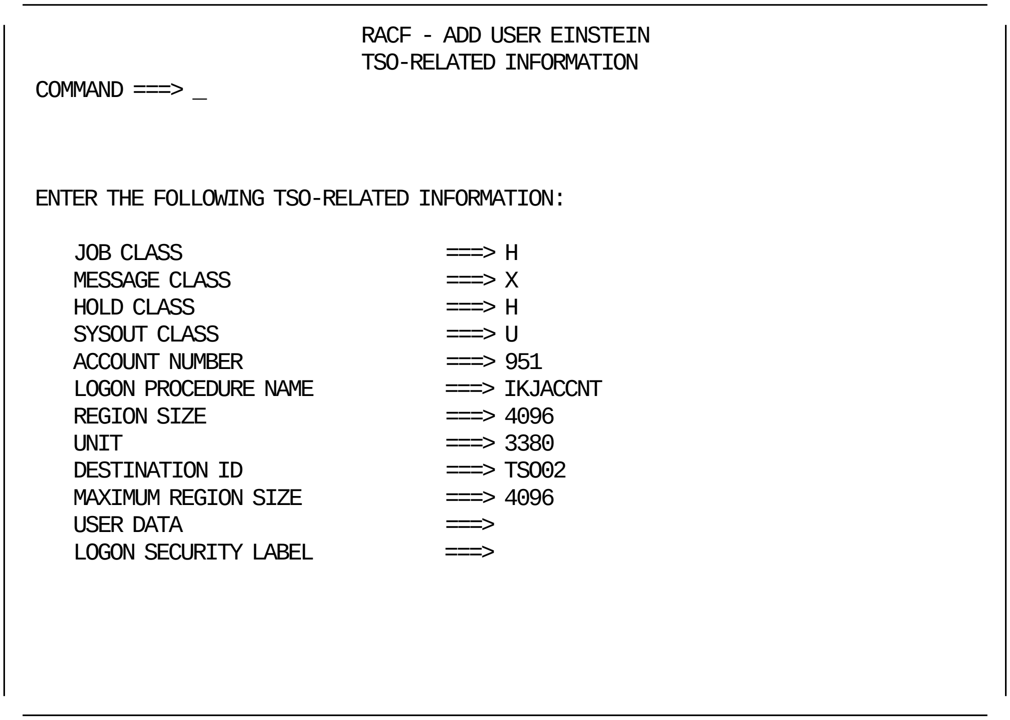
Figure 49. Adding New Users - Panel 2



```
                          RACF - ADD USER EINSTEIN
COMMAND ===> _



To ADD the following information, enter any character:

  _ CLASS AUTHORITY
  _ INSTALLATION DATA
  _ GROUP AUTHORITY
  _ SECURITY LEVEL or CATEGORIES
  _ SECURITY LABEL
  _ LOGON RESTRICTIONS
  _ NATIONAL LANGUAGES

  _ DFP PARAMETERS
  X TSO PARAMETERS
  _ OPERPARM PARAMETERS
  _ CICS PARAMETERS
```

Figure 50. Adding New Users - Panel 3

Figure 51 shows an example of the TSO information to user EINSTEIN.



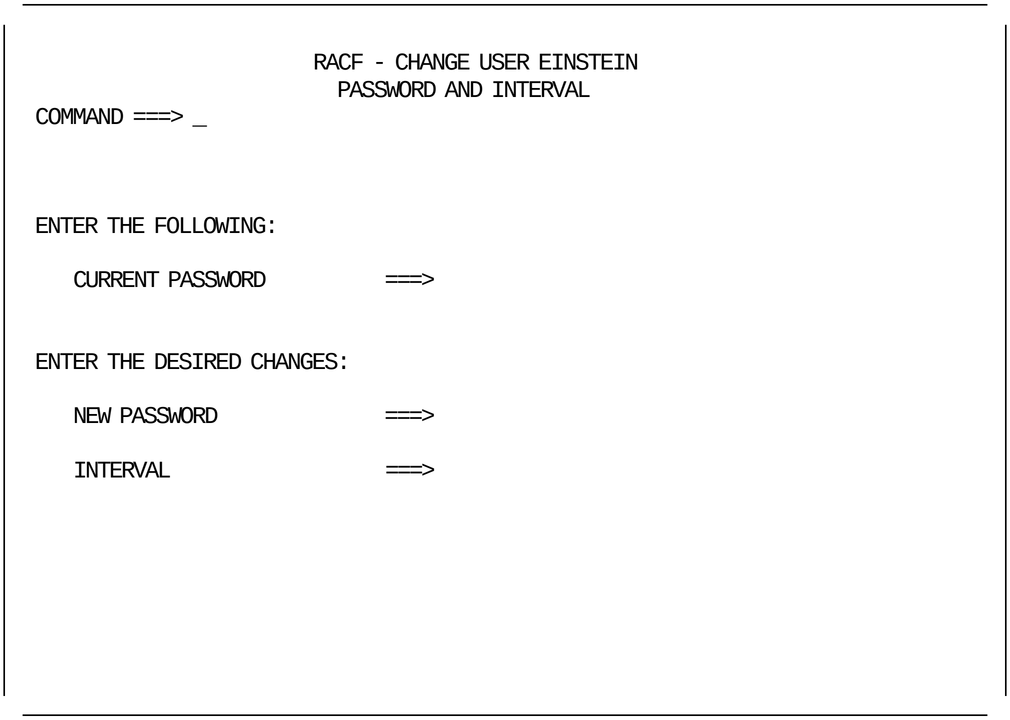
```
                           RACF - ADD USER EINSTEIN
                           TSO-RELATED INFORMATION
COMMAND ===> _



ENTER THE FOLLOWING TSO-RELATED INFORMATION:

   JOB CLASS                    ===> H
   MESSAGE CLASS                ===> X
   HOLD CLASS                   ===> H
   SYSOUT CLASS                 ===> U
   ACCOUNT NUMBER               ===> 951
   LOGON PROCEDURE NAME         ===> IKJACCNT
   REGION SIZE                  ===> 4096
   UNIT                         ===> 3380
   DESTINATION ID               ===> TSO02
   MAXIMUM REGION SIZE          ===> 4096
   USER DATA                    ===>
   LOGON SECURITY LABEL         ===>
```

Figure 51. Adding New Users - Panel 4

9.4.2 Changing Password or Interval

The fourth option of the RACF User Profile Services panel is used to change your own password or to change the password interval. You can make changes only to your own user ID, even if you have specified another user on the previous panel.

The current and new password will not appear on the panel when you enter them.



```
                         RACF - CHANGE USER EINSTEIN
                           PASSWORD AND INTERVAL
COMMAND ===> _



ENTER THE FOLLOWING:

   CURRENT PASSWORD          ===>


ENTER THE DESIRED CHANGES:

   NEW PASSWORD              ===>

   INTERVAL                  ===>
```

Figure 52. Changing Password and Interval

9.4.3 Monitoring User Activities

Option 5 of the RACF User Profile Services panel allows you to monitor user activities in the system. If you specify YES, the system logs all RACF-related activities for a specific user. For more details, see Section 8.1.8, “Auditing Considerations” on page 85.

9.4.4 Displaying User Profile

Through option 8 of the RACF User Profile Service you can list information about users. Figure 53 is the output listing for the user ID EINSTEIN, including the TSO information.

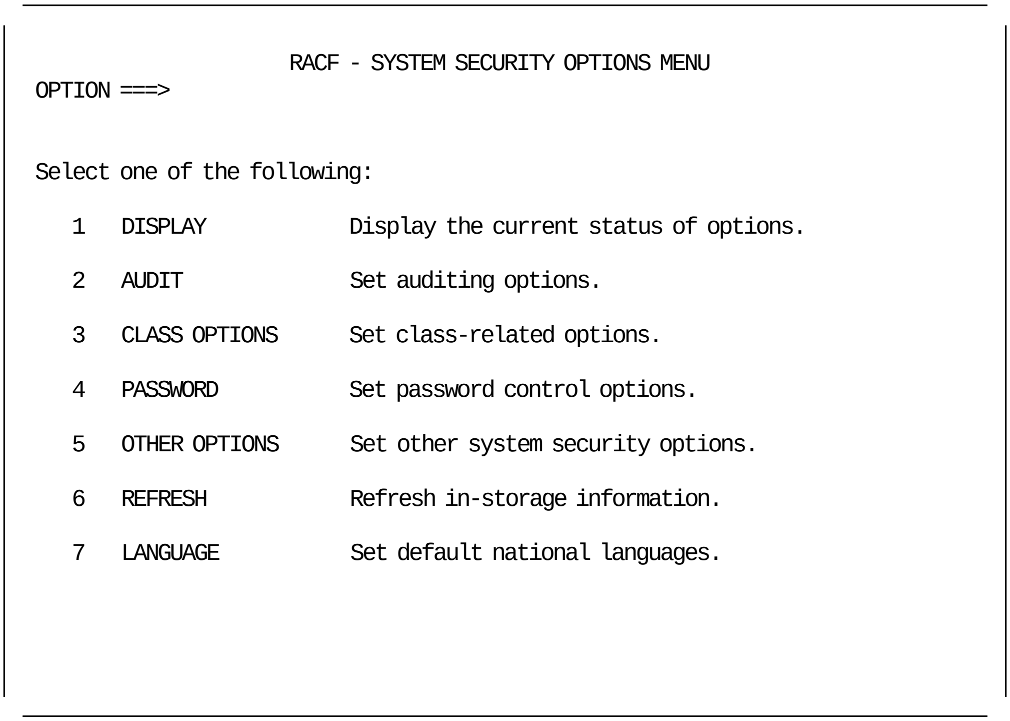
```
USER=EINSTEIN  NAME=ALBERT EINSTEIN       OWNER=RACFADM   CREATED=92.100
 DEFAULT-GROUP=GADM      PASSDATE=00.000  PASS-INTERVAL=30
 ATTRIBUTES=NONE
 REVOKE DATE=NONE   RESUME DATE=NONE
 LAST-ACCESS=UNKNOWN
 CLASS AUTHORIZATIONS=NONE
 NO-INSTALLATION-DATA
 NO-MODEL-NAME
 LOGON ALLOWED   (DAYS)          (TIME)
 ---------------------------------------------
 ANYDAY                          ANYTIME
  GROUP=GADM      AUTH=USE      CONNECT-OWNER=RACFADM   CONNECT-DATE=92.100
    CONNECTS=    00  UACC=NONE     LAST-CONNECT=UNKNOWN
    CONNECT ATTRIBUTES=NONE
    REVOKE DATE=NONE   RESUME DATE=NONE
SECURITY-LEVEL=NONE SPECIFIED
CATEGORY-AUTHORIZATION
 NONE SPECIFIED
SECURITY-LABEL=NONE SPECIFIED

TSO INFORMATION
---------------
ACCTNUM= 951
HOLDCLASS= H
JOBCLASS= H
MSGCLASS= X
PROC= IKJACCNT
SIZE= 00004096
MAXSIZE= 00004096
SYSOUTCLASS= U
UNIT= 3380
USERDATA= 0000
```

Figure 53. User Profile Display

9.5 System Options

When you select option 5 from the RACF Service Options menu, the System Security Options menu is displayed on your screen (Figure 54).

This option allows you to display, modify, and refresh the system options. This panel is similar to the SETROPTS command. To use this option you must have RACF authority - usually Auditor or Special authority.



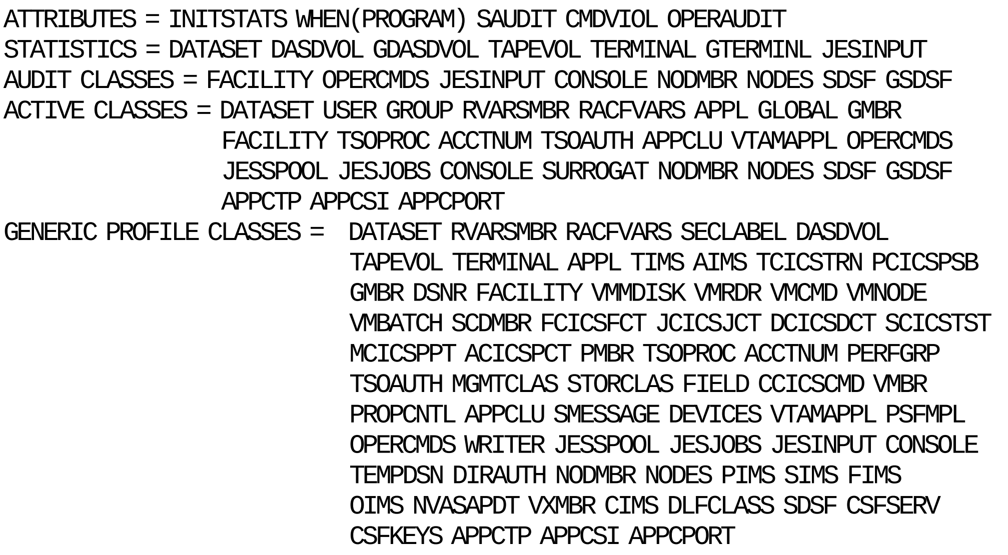
```
                      RACF - SYSTEM SECURITY OPTIONS MENU
OPTION ===>


Select one of the following:

   1   DISPLAY            Display the current status of options.

   2   AUDIT              Set auditing options.

   3   CLASS OPTIONS      Set class-related options.

   4   PASSWORD           Set password control options.

   5   OTHER OPTIONS      Set other system security options.

   6   REFRESH            Refresh in-storage information.

   7   LANGUAGE           Set default national languages.
```

Figure 54. System Security Options Menu

9.5.1 Displaying the Current Status of Options

Option 1 in the RACF System Security Options Menu displays the RACF options, such as the SETROPTS LIST command. To use this option, you must have Auditor or Special authority.

Figure 55 is an example of the current status of the RACF options.



```
ATTRIBUTES = INITSTATS WHEN(PROGRAM) SAUDIT CMDVIOL OPERAUDIT
STATISTICS = DATASET DASDVOL GDASDVOL TAPEVOL TERMINAL GTERMINL JESINPUT
AUDIT CLASSES = FACILITY OPERCMDS JESINPUT CONSOLE NODMBR NODES SDSF GSDSF
ACTIVE CLASSES = DATASET USER GROUP RVARSMBR RACFVARS APPL GLOBAL GMBR
                 FACILITY TSOPROC ACCTNUM TSOAUTH APPCLU VTAMAPPL OPERCMDS
                 JESSPOOL JESJOBS CONSOLE SURROGAT NODMBR NODES SDSF GSDSF
                 APPCTP APPCSI APPCPORT
GENERIC PROFILE CLASSES =  DATASET RVARSMBR RACFVARS SECLABEL DASDVOL
                           TAPEVOL TERMINAL APPL TIMS AIMS TCICSTRN PCICSPSB
                           GMBR DSNR FACILITY VMMDISK VMRDR VMCMD VMNODE
                           VMBATCH SCDMBR FCICSFCT JCICSJCT DCICSDCT SCICSTST
                           MCICSPPT ACICSPCT PMBR TSOPROC ACCTNUM PERFGRP
                           TSOAUTH MGMTCLAS STORCLAS FIELD CCICSCMD VMBR
                           PROPCNTL APPCLU SMESSAGE DEVICES VTAMAPPL PSFMPL
                           OPERCMDS WRITER JESSPOOL JESJOBS JESINPUT CONSOLE
                           TEMPDSN DIRAUTH NODMBR NODES PIMS SIMS FIMS
                           OIMS NVASAPDT VXMBR CIMS DLFCLASS SDSF CSFSERV
                           CSFKEYS APPCTP APPCSI APPCPORT
```

Figure 55 (Part 1 of 3). RACF Current Options Status

```
GENERIC COMMAND CLASSES =  DATASET RVARSMBR RACFVARS SECLABEL DASDVOL
                           TAPEVOL TERMINAL APPL TIMS AIMS TCICSTRN PCICSPSB
                           GMBR DSNR FACILITY VMMDISK VMRDR VMCMD VMNODE
                           VMBATCH SCDMBR FCICSFCT JCICSJCT DCICSDCT SCICSTST
                           MCICSPPT ACICSPCT PMBR TSOPROC ACCTNUM PERFGRP
                           TSOAUTH MGMTCLAS STORCLAS FIELD CCICSCMD VMBR
                           PROPCNTL APPCLU SMESSAGE DEVICES VTAMAPPL PSFMPL
                           OPERCMDS WRITER JESSPOOL JESJOBS JESINPUT CONSOLE
                           TEMPDSN DIRAUTH NODMBR NODES PIMS SIMS FIMS
                           OIMS NVASAPDT VXMBR CIMS DLFCLASS SDSF CSFSERV
                           CSFKEYS APPCTP APPCSI APPCPORT
GENLIST CLASSES =  DASDVOL JESJOBS
GLOBAL CHECKING CLASSES =  RVARSMBR SECLABEL DASDVOL TAPEVOL TERMINAL
                           APPL TIMS AIMS TCICSTRN PCICSPSB GMBR DSNR
                           FACILITY VMMDISK VMRDR VMCMD VMNODE VMBATCH
                           SCDMBR FCICSFCT JCICSJCT DCICSDCT SCICSTST
                           MCICSPPT ACICSPCT PMBR TSOPROC ACCTNUM PERFGRP
                           TSOAUTH MGMTCLAS STORCLAS FIELD CCICSCMD VMBR
                           PROPCNTL APPCLU SMESSAGE DEVICES VTAMAPPL PSFMPL
                           OPERCMDS WRITER JESSPOOL JESJOBS JESINPUT CONSOLE
                           TEMPDSN DIRAUTH NODMBR PIMS SIMS FIMS OIMS
                           NVASAPDT VXMBR CIMS DLFCLASS SDSF
RACLIST CLASSES =  RACFVARS APPL FACILITY TSOAUTH VTAMAPPL OPERCMDS JESSPOOL
                   CONSOLE SURROGAT NODES SDSF APPCTP APPCSI APPCPORT
LOGOPTIONS "ALWAYS" CLASSES =  WRITER JESSPOOL JESJOBS JESINPUT SURROGAT
                               NODMBR NODES
LOGOPTIONS "NEVER" CLASSES =  NONE
LOGOPTIONS "SUCCESSES" CLASSES =  NONE
LOGOPTIONS "FAILURES" CLASSES =  OPERCMDS CONSOLE
LOGOPTIONS "DEFAULT" CLASSES =  DATASET RVARSMBR RACFVARS SECLABEL DASDVOL
                                GDASDVOL TAPEVOL TERMINAL GTERMINL APPL
                                TIMS GIMS AIMS TCICSTRN GCICSTRN PCICSPSB
                                QCICSPSB GLOBAL GMBR DSNR FACILITY VMMDISK
                                VMRDR VMCMD VMNODE VMBATCH SCDMBR SECDATA
                                FCICSFCT HCICSFCT JCICSJCT KCICSJCT DCICSDCT
                                ECICSDCT SCICSTST UCICSTST MCICSPPT NCICSPPT
                                ACICSPCT BCICSPCT PMBR PROGRAM TSOPROC
                                ACCTNUM PERFGRP TSOAUTH MGMTCLAS STORCLAS
                                FIELD CCICSCMD VCICSCMD VMBR VMEVENT PROPCNTL
                                APPCLU SMESSAGE DEVICES VTAMAPPL PSFMPL
                                TEMPDSN DIRAUTH PIMS QIMS SIMS UIMS FIMS
                                HIMS OIMS WIMS NVASAPDT VXMBR VMXEVENT
                                CIMS DIMS DLFCLASS SDSF GSDSF CSFSERV
                                CSFKEYS GCSFKEYS APPCTP APPCSI APPCPORT
                                RMTOPS
AUTOMATIC DATASET PROTECTION IS IN EFFECT
ENHANCED GENERIC NAMING IS IN EFFECT
REAL DATA SET NAMES OPTION IS INACTIVE
JES-BATCHALLRACF OPTION IS ACTIVE
JES-XBMALLRACF OPTION IS INACTIVE
JES-EARLYVERIFY OPTION IS INACTIVE
PROTECT-ALL OPTION IS NOT IN EFFECT
TAPE DATA SET PROTECTION IS INACTIVE
SECURITY RETENTION PERIOD IN EFFECT IS     0 DAYS.
ERASE-ON-SCRATCH IS INACTIVE
```

Figure 55 (Part 2 of 3). RACF Current Options Status

```
SINGLE LEVEL NAME PREFIX IS PASSWORD
LIST OF GROUPS ACCESS CHECKING IS ACTIVE.
INACTIVE USERIDS ARE NOT BEING AUTOMATICALLY REVOKED.
NO DATA SET MODELLING BEING DONE.
PASSWORD PROCESSING OPTIONS:
  PASSWORD CHANGE INTERVAL IS  30 DAYS.
  12 GENERATIONS OF PREVIOUS PASSWORDS BEING MAINTAINED.
  AFTER   6 CONSECUTIVE UNSUCCESSFUL PASSWORD ATTEMPTS,
      A USERID WILL BE REVOKED.
  PASSWORD EXPIRATION WARNING LEVEL IS   5 DAYS.
  INSTALLATION PASSWORD SYNTAX RULES:
    RULE 1  LENGTH(4:8)   CWWWWWWW
    RULE 2  LENGTH(4:8)   WCWWWWWW
    RULE 3  LENGTH(4:8)   WWCWWWWW
    RULE 4  LENGTH(4:8)   VVVNNNNN
    RULE 5  LENGTH(5:8)   CCCCCWWW
    RULE 6  LENGTH(6:8)   NNNNNNWW
   LEGEND:
    A-ALPHA C-CONSONANT L-ALPHANUM N-NUMERIC V-VOWEL W-NOVOWEL *-ANYTHING
DEFAULT RVARY PASSWORD IS IN EFFECT FOR THE SWITCH FUNCTION.
DEFAULT RVARY PASSWORD IS IN EFFECT FOR THE STATUS FUNCTION.
SECLEVELAUDIT IS INACTIVE
SECLABEL AUDIT IS IN EFFECT
SECLABEL CONTROL IS NOT IN EFFECT
GENERIC OWNER ONLY IS NOT IN EFFECT
COMPATIBILITY MODE IS NOT IN EFFECT
MULTI-LEVEL QUIET IS NOT IN EFFECT
MULTI-LEVEL STABLE IS NOT IN EFFECT
MULTI-LEVEL SECURE IS NOT IN EFFECT
MULTI-LEVEL ACTIVE IS NOT IN EFFECT
CATALOGUED DATA SETS ONLY, IS NOT IN EFFECT
USER-ID FOR JES NJEUSERID IS : DFLT-NJE
USER-ID FOR JES UNDEFINEDUSER IS : UKWN-UID
PARTNER LU-VERIFICATION SESSIONKEY INTERVAL DEFAULT IS    30 DAYS.
PRIMARY LANGUAGE DEFAULT : ENU
SECONDARY LANGUAGE DEFAULT : ENU
```

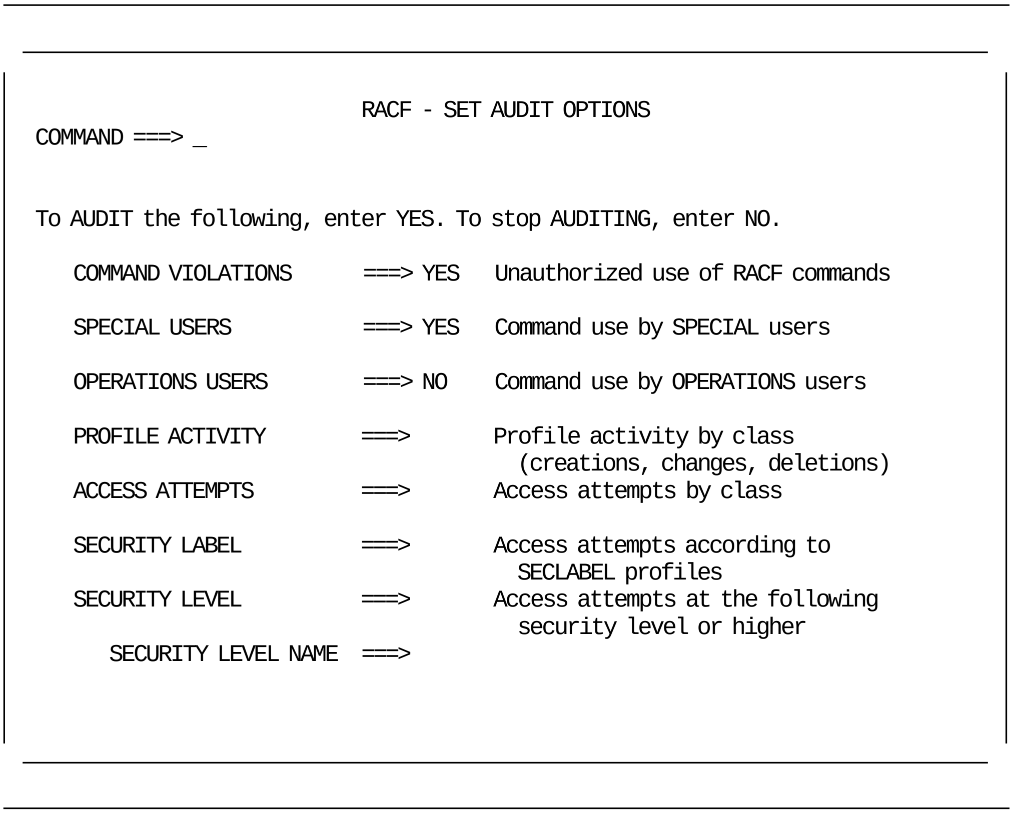
Figure 55 (Part 3 of 3). RACF Current Options Status

9.5.2 Setting Auditing Options

Option 2 in the RACF System Security Options menu sets the audit options of the system, that is, defines what is going to be logged in the SMF. You must have Auditor authority to use this option.

For example, specifying the values shown in Figure 56 has the same effect of entering the following RACF commands:

- SETROPTS CMDVIOL
- SETROPTS SAUDIT
- SETROPTS NOOPERAUDIT



```
                             RACF - SET AUDIT OPTIONS
COMMAND ===> _


To AUDIT the following, enter YES. To stop AUDITING, enter NO.

   COMMAND VIOLATIONS       ===> YES   Unauthorized use of RACF commands

   SPECIAL USERS            ===> YES   Command use by SPECIAL users

   OPERATIONS USERS         ===> NO    Command use by OPERATIONS users

   PROFILE ACTIVITY         ===>       Profile activity by class
                                         (creations, changes, deletions)
   ACCESS ATTEMPTS          ===>       Access attempts by class

   SECURITY LABEL           ===>       Access attempts according to
                                         SECLABEL profiles
   SECURITY LEVEL           ===>       Access attempts at the following
                                         security level or higher
      SECURITY LEVEL NAME   ===>
```

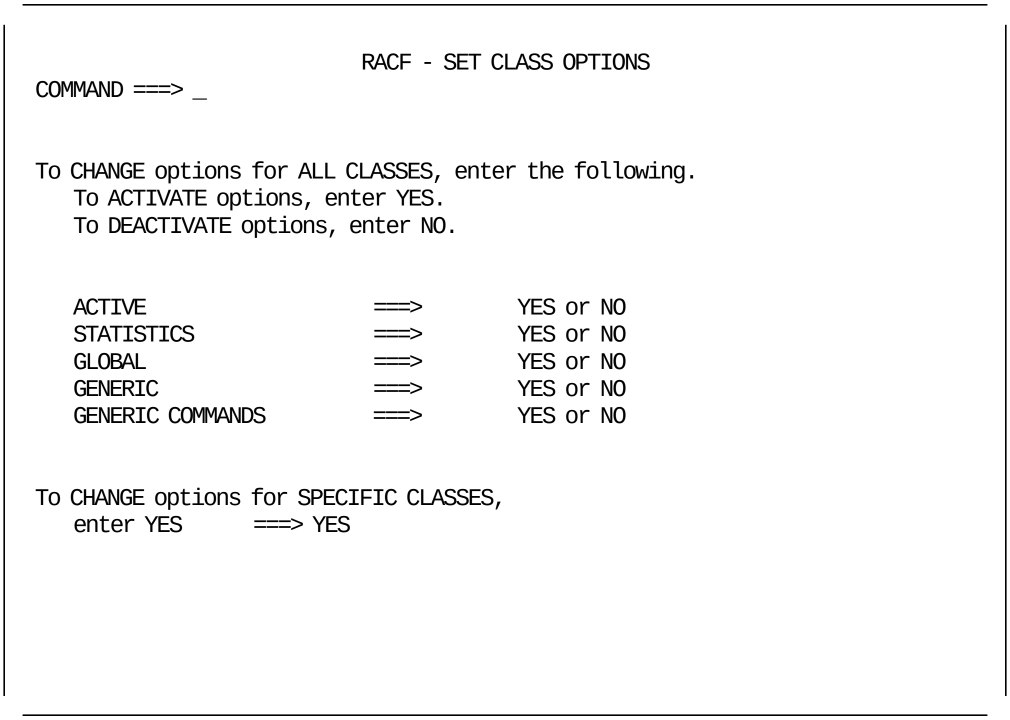
Figure 56. Setting Audit Options

9.5.3 Setting Class Options

Option 3 in the RACF System Security Options Menu sets the class options of the system. You must have Special authority to use this option.

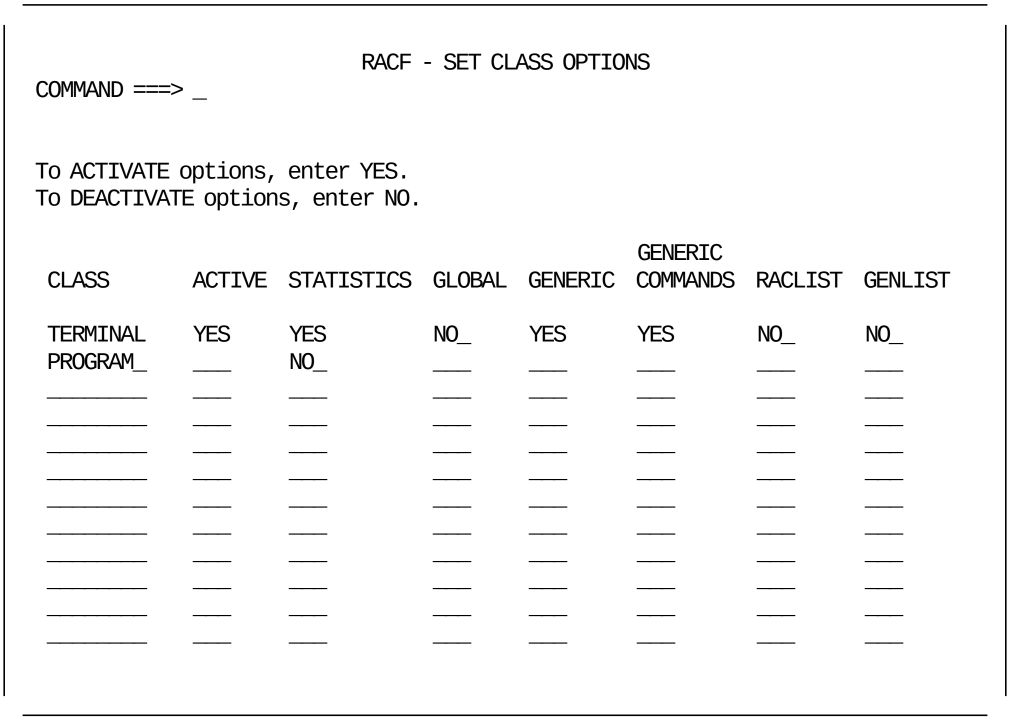
You can change the status of all the classes at one time or change the status of specific classes. It is recommended that you change the options for each class separately, as specified in Figure 57.

Then you have to enter the options for each class separately on a panel shown in Figure 58. In the example we are activating and defining options for the TERMINAL class, and turning off the statistics for the PROGRAM class, which is already defined.



```
                            RACF - SET CLASS OPTIONS
COMMAND ===> _


To CHANGE options for ALL CLASSES, enter the following.
   To ACTIVATE options, enter YES.
   To DEACTIVATE options, enter NO.


   ACTIVE                  ===>        YES or NO
   STATISTICS              ===>        YES or NO
   GLOBAL                  ===>        YES or NO
   GENERIC                 ===>        YES or NO
   GENERIC COMMANDS        ===>        YES or NO


To CHANGE options for SPECIFIC CLASSES,
   enter YES        ===> YES
```

Figure 57. Setting Class Options - Panel 1



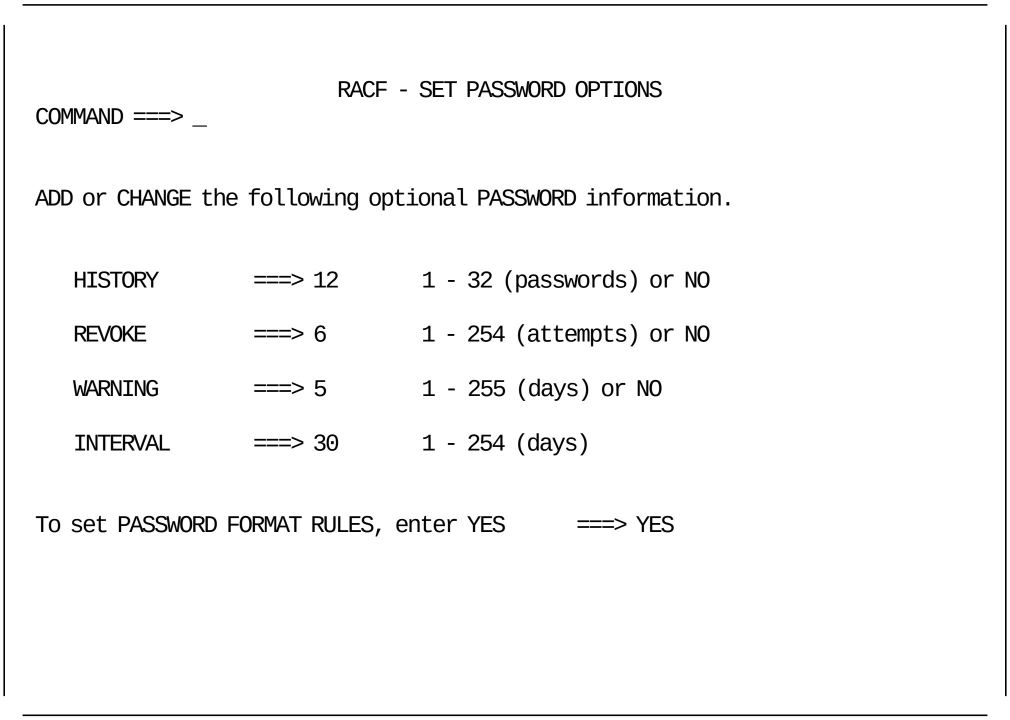
```
                            RACF - SET CLASS OPTIONS
COMMAND ===> _


To ACTIVATE options, enter YES.
To DEACTIVATE options, enter NO.

                                                     GENERIC
 CLASS       ACTIVE  STATISTICS  GLOBAL  GENERIC  COMMANDS  RACLIST  GENLIST

 TERMINAL    YES     YES         NO_     YES      YES       NO_      NO_
 PROGRAM_    ___     NO_         ___     ___      ___       ___      ___
 ________    ___     ___         ___     ___      ___       ___      ___
 ________    ___     ___         ___     ___      ___       ___      ___
 ________    ___     ___         ___     ___      ___       ___      ___
 ________    ___     ___         ___     ___      ___       ___      ___
 ________    ___     ___         ___     ___      ___       ___      ___
 ________    ___     ___         ___     ___      ___       ___      ___
 ________    ___     ___         ___     ___      ___       ___      ___
 ________    ___     ___         ___     ___      ___       ___      ___
 ________    ___     ___         ___     ___      ___       ___      ___
 ________    ___     ___         ___     ___      ___       ___      ___
```

Figure 58. Setting Class Options - Panel 2

9.5.4 Setting Password Options

Option 4 in the RACF System Security Options menu sets the password options.

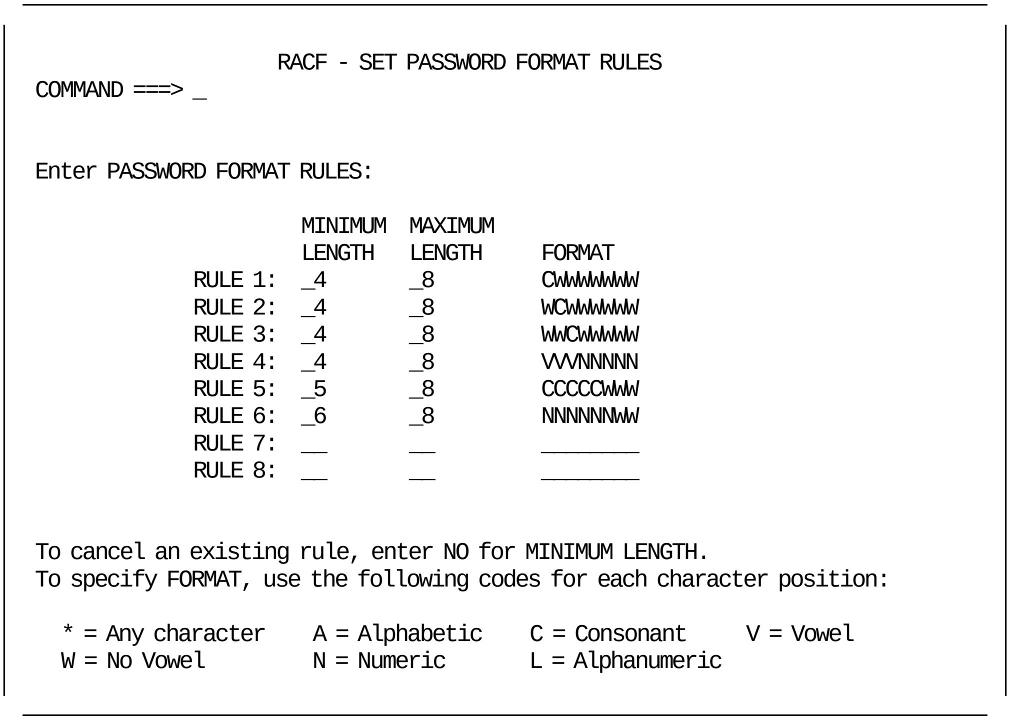
Figure 59 is an example of password options. It is recommended that you specify password rules to avoid the use of trivial passwords.



```
                         RACF - SET PASSWORD OPTIONS
COMMAND ===> _


ADD or CHANGE the following optional PASSWORD information.


   HISTORY       ===> 12      1 - 32 (passwords) or NO

   REVOKE        ===> 6       1 - 254 (attempts) or NO

   WARNING       ===> 5       1 - 255 (days) or NO

   INTERVAL      ===> 30      1 - 254 (days)

To set PASSWORD FORMAT RULES, enter YES     ===> YES
```

Figure 59. Setting Password Options

Figure 60 shows an example of password rules. When a user changes a password, the new password has to match one of the defined rules. In the example, a user cannot use the password XWZ because it has a length less than four. But the user can specify, for example, 245436, BCD32, or AEI42 as a new password.



```
                    RACF - SET PASSWORD FORMAT RULES
COMMAND ===> _


Enter PASSWORD FORMAT RULES:

                       MINIMUM  MAXIMUM
                       LENGTH   LENGTH     FORMAT
             RULE 1:   _4       _8         CWWWWWWW
             RULE 2:   _4       _8         WCWWWWWW
             RULE 3:   _4       _8         WWCWWWWW
             RULE 4:   _4       _8         VVVNNNNN
             RULE 5:   _5       _8         CCCCCWWW
             RULE 6:   _6       _8         NNNNNNWW
             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     A = Alphabetic     C = Consonant     V = Vowel
  W = No Vowel          N = Numeric        L = Alphanumeric
```

Figure 60. Setting Password Rules

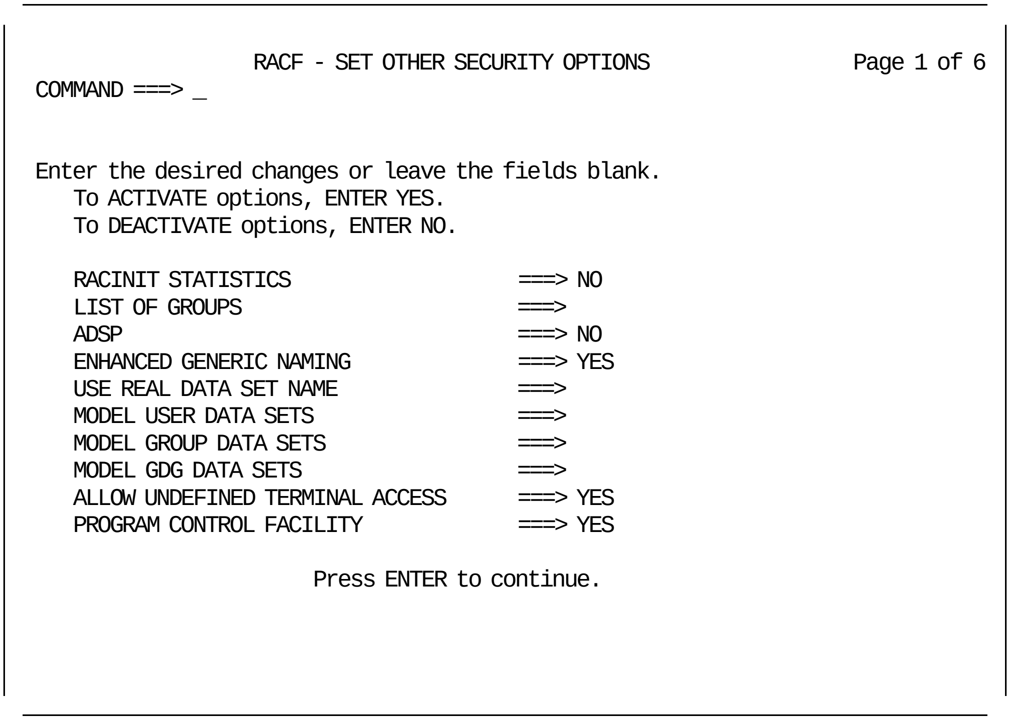
9.5.5 Setting Other Security Options

Option 5 in the RACF System Security Options menu sets several security options.

Figure 61 shows an example of turning off the STATISTICS and ADSP options, and turning on the EGN, TERMINAL, and the WHEN(PROGRAM) options.

These options have the following effects:

- RACF will not record statistics of accesses to resources protected by discrete profiles (NOSTATISTICS).
- RACF will not force the automatic creation of profiles (NOADSP).
- Enhanced generic naming will be active (EGN).
- Users will be able to log on to the system using undefined terminals (TERMINAL).
- RACF will construct a table that describes the controlled programs and who can access them (WHEN(PROGRAM)).



```
                  RACF - SET OTHER SECURITY OPTIONS                Page 1 of 6
COMMAND ===> _


Enter the desired changes or leave the fields blank.
   To ACTIVATE options, ENTER YES.
   To DEACTIVATE options, ENTER NO.

   RACINIT STATISTICS                  ===> NO
   LIST OF GROUPS                      ===>
   ADSP                                ===> NO
   ENHANCED GENERIC NAMING             ===> YES
   USE REAL DATA SET NAME              ===>
   MODEL USER DATA SETS                ===>
   MODEL GROUP DATA SETS               ===>
   MODEL GDG DATA SETS                 ===>
   ALLOW UNDEFINED TERMINAL ACCESS     ===> YES
   PROGRAM CONTROL FACILITY            ===> YES

                       Press ENTER to continue.
```

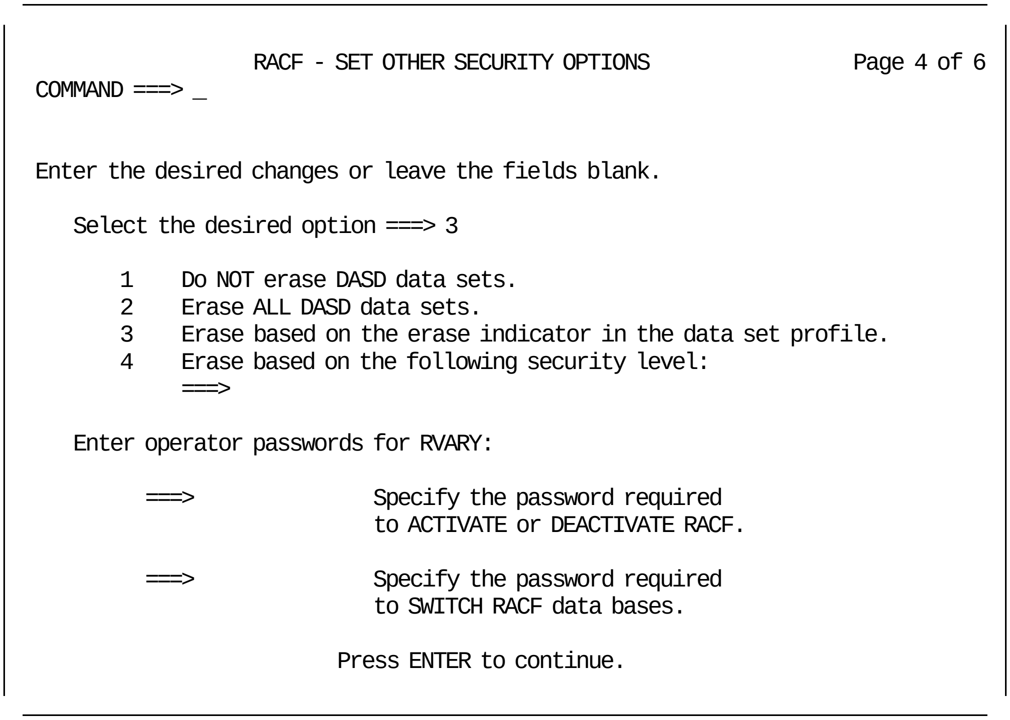
Figure 61. Setting Security Options - Panel 1

Figure 62 shows an example of turning on the Erase-on-scratch option only when the ERASE indicator is specified in the data set profile. We recommend this approach rather than erasing all data sets, since the erase-on-scratch places an additional load on DASDs, and impacts MVS system performance.

Using this screen, you can also define the RVARY SWITCH and STATUS passwords.

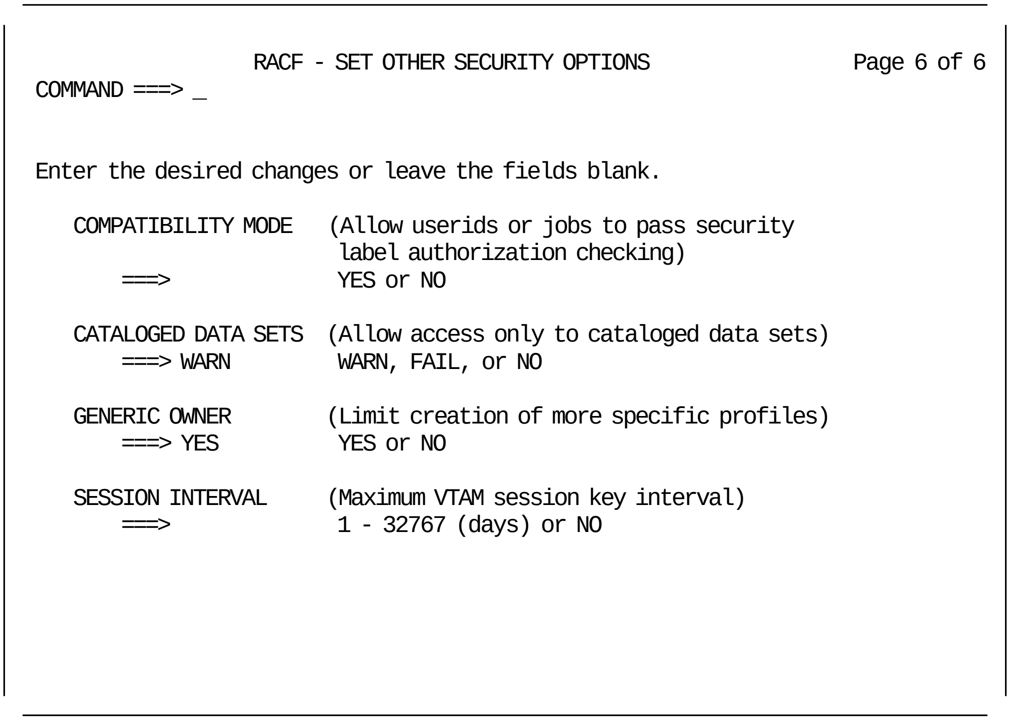
Figure 63 shows an example of specifying the CATDSNS option with the WARNING operand. This option allows an authorized user to access uncataloged data sets, but sends a warning message to this user and the security administrator.

Also, the GENERICOWNER option is activated, restricting the creation of profiles in general resource classes to only specific users.



```
                  RACF - SET OTHER SECURITY OPTIONS                Page 4 of 6
COMMAND ===> _


Enter the desired changes or leave the fields blank.

   Select the desired option ===> 3

       1    Do NOT erase DASD data sets.
       2    Erase ALL DASD data sets.
       3    Erase based on the erase indicator in the data set profile.
       4    Erase based on the following security level:
            ===>

   Enter operator passwords for RVARY:

         ===>               Specify the password required
                            to ACTIVATE or DEACTIVATE RACF.

         ===>               Specify the password required
                            to SWITCH RACF data bases.

                         Press ENTER to continue.
```

Figure 62. Setting Security Options - Panel 2



```
                  RACF - SET OTHER SECURITY OPTIONS                Page 6 of 6
COMMAND ===> _


Enter the desired changes or leave the fields blank.

   COMPATIBILITY MODE   (Allow userids or jobs to pass security
                         label authorization checking)
       ===>              YES or NO

   CATALOGED DATA SETS  (Allow access only to cataloged data sets)
       ===> WARN         WARN, FAIL, or NO

   GENERIC OWNER        (Limit creation of more specific profiles)
       ===> YES          YES or NO

   SESSION INTERVAL     (Maximum VTAM session key interval)
       ===>              1 - 32767 (days) or NO
```

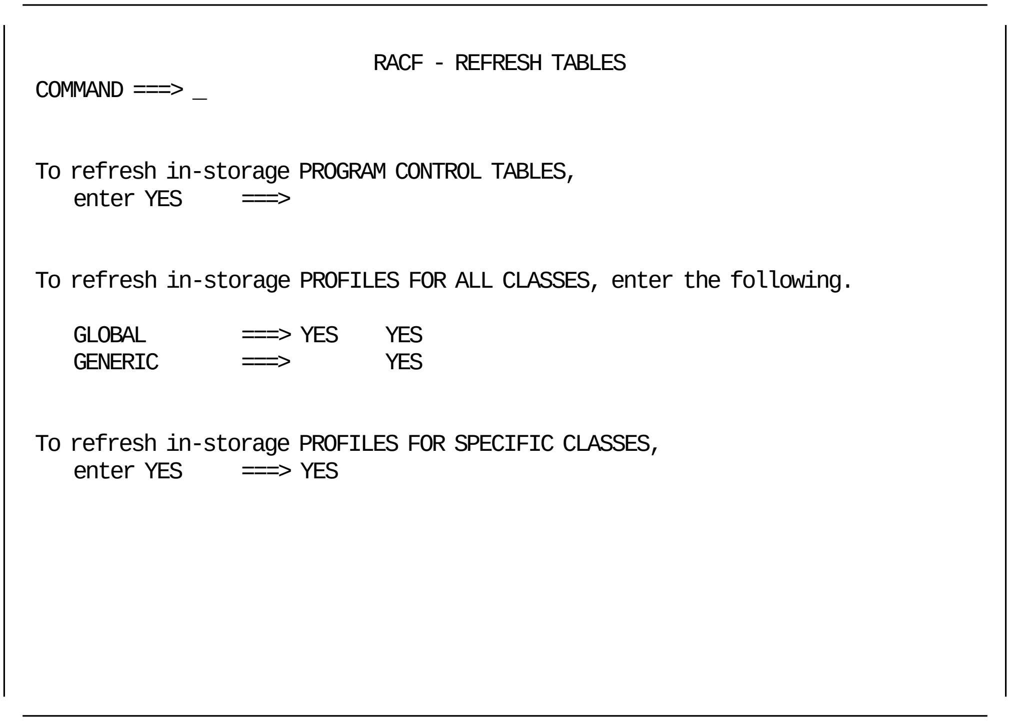
Figure 63. Setting Security Options - Panel 3

9.5.6 Refreshing In-Storage Information

Some RACF information is kept in storage. When you change this information in the RACF database you must refresh the in-storage information so that the change could affect the system. In-storage information includes classes that are defined in RACLIST.

Option 6 of the System Security Options menu allows you to refresh the in-storage information.

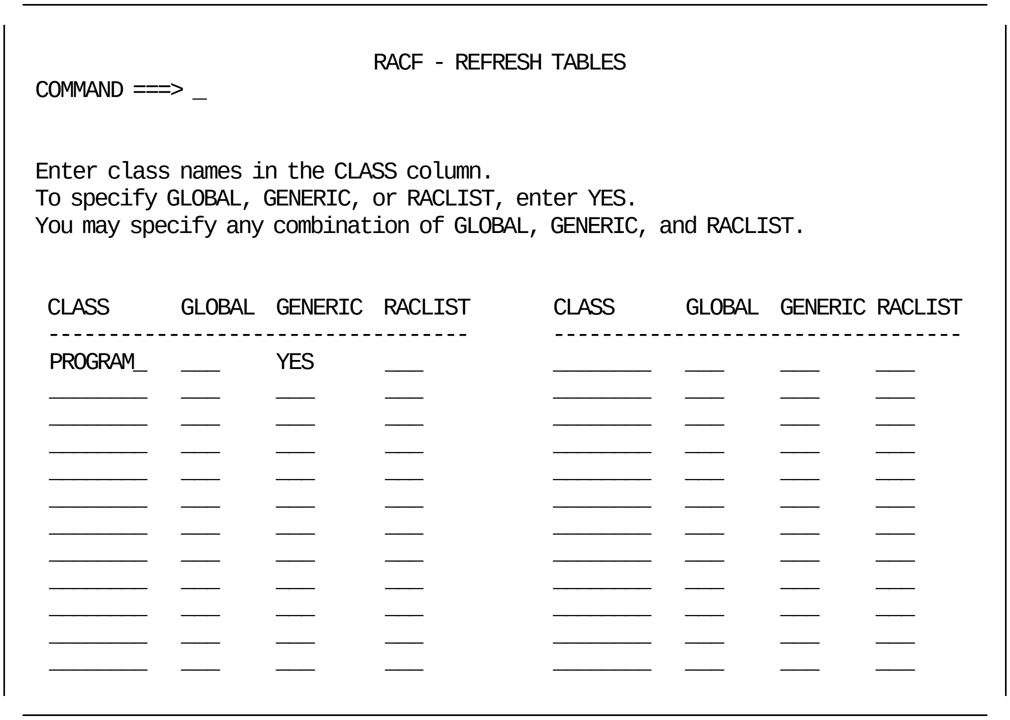
In Figure 64 the GLOBAL option is being refreshed for all classes (you have to refresh GLOBAL if you make changes in GAC table) and we are going to specify the refresh for specific classes.



```
                              RACF - REFRESH TABLES
COMMAND ===> _


To refresh in-storage PROGRAM CONTROL TABLES,
   enter YES      ===>


To refresh in-storage PROFILES FOR ALL CLASSES, enter the following.

   GLOBAL         ===> YES    YES
   GENERIC        ===>        YES


To refresh in-storage PROFILES FOR SPECIFIC CLASSES,
   enter YES      ===> YES
```

Figure 64. Refreshing In-Storage Information - Panel 1

Figure 65 shows the GENERIC option being refreshed for only the PROGRAM class.



```
                           RACF - REFRESH TABLES
COMMAND ===> _


Enter class names in the CLASS column.
To specify GLOBAL, GENERIC, or RACLIST, enter YES.
You may specify any combination of GLOBAL, GENERIC, and RACLIST.


 CLASS     GLOBAL  GENERIC  RACLIST      CLASS     GLOBAL  GENERIC RACLIST
 ---------------------------------      ---------------------------------
 PROGRAM_   ___    YES      ___          ________  ___     ___     ___
 ________   ___    ___      ___          ________  ___     ___     ___
 ________   ___    ___      ___          ________  ___     ___     ___
 ________   ___    ___      ___          ________  ___     ___     ___
 ________   ___    ___      ___          ________  ___     ___     ___
 ________   ___    ___      ___          ________  ___     ___     ___
 ________   ___    ___      ___          ________  ___     ___     ___
 ________   ___    ___      ___          ________  ___     ___     ___
 ________   ___    ___      ___          ________  ___     ___     ___
 ________   ___    ___      ___          ________  ___     ___     ___
 ________   ___    ___      ___          ________  ___     ___     ___
 ________   ___    ___      ___          ________  ___     ___     ___
```

Figure 65. Refreshing In-Storage Information - Panel 2

Chapter 10. MVS Integrity

Data security is the protection of data against unauthorized disclosure, transfer, modifications, or destruction, whether accidental or intentional. Data security involves more than the security software. There are some features in MVS that must be controlled, since they can be used to bypass the security provided by RACF.

An operating system should have integrity. It should be designed, implemented and maintained to prevent unauthorized access that can compromise the system. MVS integrity should prevent any unauthorized program, using any defined or undefined system interface, from:

- Bypassing store or fetch protection. For example, reading or writing into other user′s area.
- Bypassing OS password, VSAM password or RACF security checking. For example, accessing a password-protected data without supplying the correct password.
- Obtaining control in an authorized state. For example, running as an authorized program with privileges (APF authorized, or in a system key, or in supervisor state).

In spite of security controls managed by MVS, the security administrator must ensure that installation modification and additions to the control program do not introduce any integrity exposures. That is, all installation-written authorized code (such as an installation SVC) must perform the same or an equivalent type of validity checking and control that the system uses to maintain its integrity. The installation is also responsible for adopting certain procedures that are necessary to complement the integrity enhancement within the operating system itself.

The installation security administrator can use the RACF DSMON facility to monitor the status and protection of the resources that are critical to system integrity. For more information about the use of DSMON and the reports that can be produced, refer to 8.2, “Data Security Monitor” on page 86.

10.1 Hardware Features for Integrity

To achieve overall security of systems, MVS uses some hardware architecture features such as:

- Private address spaces - In MVS, virtual storage consists of a system area, a common area, and a private area. Every MVS user can address only one private area, which is isolated and protected against other user′s access. Before MVS allocates the real storage backing this private address space to another user, it clears any residual data from it, preventing the access of its previous content.
- Multiple storage protection keys - MVS protects information in real storage from unauthorized use by means of multiple storage protection keys. The key in storage contains the protection key of the information owner, which protects the storage block from unauthorized modification; and a fetch protection bit, which protects the block of storage from an unauthorized attempt to read or fetch its contents. When unauthorized attempts to read or

modify the contents of real storage are detected, a program exception interrupt is issued.

- Channel program translation - To protect programs from reading or writing information from or into system′s or other user′s main storage areas, MVS translates all user-written channel programs. That is, MVS replaces the channel program′s virtual storage addresses with real addresses and builds a new, protected copy of the channel program. MVS also prefixes user-written channel programs with a Set File Mask command which prevents the channel program from reading or writing from or into another user′s data set extents.

10.2 Authorized programs

Many system functions, such as supervisor calls (SVC) or special paths through SVCs, are sensitive; that is, access to these functions must be restricted to authorized programs to avoid compromising the security and integrity of the systems.

Only authorized programs can access restricted SVCs in MVS. A program is authorized if it satisfies at least one of the following conditions:

- The program is executing with, or with a capability to obtain, a PSW that contains a system protection key (bits 8-11 of the PSW are in the range of 0-7; in other words, bit 8 is zero.)
- The program is executing with a PSW that is in supervisor state (bit 15 of the PSW is 0)
- The program is executing with Authorized Program Facility (APF) authorization (as part of an APF-authorized job step task).

10.2.1 Authorized Program Facility (APF)

The Authorized Program Facility (APF) is the primary mechanism for security and control within the MVS operating system. APF authorized programs are considered an extension of the operating system, and can bypass all security mechanisms.

The authorized program facility allows the installation to identify system programs and user programs that are permitted to use sensitive system functions.

The system uses APF as follows:

- It limits the use of sensitive SVC routines and optionally, sensitive user SVC routines, to authorized programs.
- It ensures that all modules in an authorized job step task reside only in authorized libraries. This check prevents the unauthorized counterfeiting of any module in an authorized job step task′s module flow.

10.2.1.1 APF Authorization

APF also enables a program to become authorized. At the time the task is initiated, the authorization status of the job step task′s first module determines the APF authorization. The job step task is marked as APF authorized only when the first module loaded comes from an authorized library, and was link-edited with the authorization code AC=1. If the first module is marked as authorized, the system assumes that the flow of control to all subsequent modules is known and secure, as long as these subsequent modules come from authorized libraries (if they do not, an abend results).

To restrict the submission of a program by an individual user or a class of users, you can place the program in a library, other than LINKLIB, SVCLIB or LPALIB, protected by RACF or by a password. If the program is an authorized program, the library must be an installation authorized library. An authorized program can also be restricted by defining it as a RACF resource using the RACROUTE REQUEST=AUTH macro instruction in the program to verify the user′s authorization or by using the RACF PROGRAM class for access authorization.

10.2.1.2 Authorized Libraries

APF-authorized programs must reside in authorized libraries, that are the following:

- SYS1.LINKLIB
- SYS1.SVCLIB
- SYS1.LPALIB (only during an IPL)
- Installation-defined authorized libraries

To allow an installation to authorize libraries, use member IEAAPFxx in SYS1.PARMLIB. SYS1.LINKLIB and SYS1.SVCLIB are automatically made authorized libraries. The LNKLSTxx members of SYS1.PARMLIB denote libraries to be concatenated to SYS1.LINKLIB.

10.2.1.3 Controlling the Integrity

Installations using APF authorization must control which programs are stored in authorized libraries to ensure the system integrity. The installation should:

- Ensure that all programs that run as authorized programs adhere to the installation′s integrity guidelines and to the MVS system integrity requirements.
- Control which programs are stored in authorized libraries.
- Protect authorized libraries through RACF or passwords to ensure that only selected users can store programs in these libraries.
- Ensure that no two load modules with the same name exist across the set of authorized libraries. Two modules with the same name can lead to accidental or deliberate mix-up in module flow, possibly introducing an integrity exposure.
- Link-edit with the authorization code AC=1 only the first load module in a program sequence. Do not use the authorization code for subsequent load modules; this ensures that a user cannot call modules out of sequence, become APF-authorized, and thus possibly bypass validity checking or critical logic flow.
- Ensure that IEAAPFxx does not contain the name and volume serial of data sets that no longer exist. If it does, a user can assign his own data sets with the same names on the same volumes and cause his own libraries to become authorized.
- Control who can update IEAAPFxxx. Only a few people should be able to update this SYS1.PARMLIB member.

10.2.1.4 Restricting SVC Routines

An installation can use APF to restrict the use of SVC routines to authorized programs and to restrict access to load modules so that authorized programs cannot access any load module that is not in an authorized library.

An installation can restrict the use of sensitive SVC routines in the following ways:

- To restrict the use of sensitive SVCs to authorized callers, specify the APF=YES parameter on the SVCUPDTE macro, or specify APF(YES) in the SYS1.PARMLIB member IEASVCxx.
- To restrict an entire SVC or particular paths through an SVC when only a portion of the SVC′s function is sensitive, use the TESTAUTH macro instruction.

10.2.2 Program Properties Table

For application programs to run as efficiently and securely as possible, they sometimes need to possess special properties. You can specify these special properties in the Program Properties Table (PPT) entries.

The program properties table is an MVS control block that assigns special properties to APF authorized programs. Using the entries in in the PPT, you can:

- Specify whether a program is allowed to bypass password protection or RACF (the bypass password protection (NOPASS) indicator bypasses RACF checking for data sets).
- Specify the data set integrity bypass indicator (DSI), which allows a program to allocate a data set even if it is held by another address space with DISP=OLD.
- Assign a system key (key 0-7) instead of the usual problem program key of 8.

Prior to MVS 4.1, the IBM default PPT contained a number of programs with the NOPASS option. An installation can change the IBM defaults and also add local entries to the PPT through SCHEDxx members in the SYS1.PARMLIB.

Note: Leaving extra names in the PPT makes it easier for someone to move a program with one of these names into an APF authorized library and perform unauthorized activities. Good control over the PPT requires that write access to SYS1.PARMLIB and all APF libraries are tightly controlled.

10.3 System Authorization Facility Interface

The system authorization facility (SAF) is the MVS/ESA component that provides an interface between a resource manager requesting access to a resource within the system and the security product, RACF.

Resource managers include:

- DADSM for data set access authority
- DFHSM for data set allocation authority
- CICS for CICS sign on and transaction authorization
- JES for user identification and verification

SAF is present and active on an MVS system even when RACF is not. Before MVS 3.1.3, SAF′s only function was as a router between MVS and RACF. With MVS 3.1.3, SAF establishes default security, provides security functions when RACF is not active, and performs propagation and token services.

10.3.1 MVS Router

The key element in SAF is the MVS router. The resource managing components and subsystems call the MVS router as part of certain decision-making functions in their processing, such as access control checking. SAF encourages the use of common control functions across products and systems, since it provides a focal point and a common system interface for all products providing resource control.

The RACROUTE macro is the interface to RACF (or another external security manager) for MVS resource managers. It defines the macros and keywords that are available for resource managers to implement security for data set and other resources.

MVS also provides independent RACF system macros that can be used for resource managers to invoke RACF. The use of the independent macros is not recommended because they are not being enhanced in future releases. The RACROUTE macro provides request types that are replacements for the independent RACF system macros.

Table 1 identifies the RACROUTE macro request types that replace the independent RACF system macros and their functions.

Table 1 (Page 1 of 2). Cross-reference for RACROUTE REQUEST=type and the Independent System Macros

RACROUTE REQUEST type	**Function**	**Independent macro**
REQUEST=AUTH	Provides authorization checking when a user request access to a RACF protected resource.	RACHECK
REQUEST=DEFINE	Used to define, modify, rename, or delete resource profiles for RACF.	RACDEF
REQUEST=EXTRACT	Used to retrieve or replace specified fields from a RACF resource profile or to encode data.	RACXTRT
REQUEST=FASTAUTH	Similar to REQUEST=AUTH, provides authorization checking when a user requests access to a RACF protected resource, but only for those resources whose RACF profiles have been brought into main storage by the RACROUTE REQUEST=LIST facility.	FRACHECK
REQUEST=LIST	Builds in-storage profiles for RACF defined resources for faster authorization checking.	RACLIST

Table 1 (Page 2 of 2). Cross-reference for RACROUTE REQUEST=type and the Independent System Macros

RACROUTE REQUEST type	**Function**	**Independent macro**
REQUEST=STAT	Used to determine whether RACF is active, and, optionally, to determine whether a given resource class is defined to RACF. If a resource class name is defined to RACF, the macro also determine whether the class is active.	RACSTAT
REQUEST=VERIFY	Provides RACF user identification and verification.	RACINIT

10.3.2 SAF Function

Resource managers use macro RACROUTE to call SAF MVS router to determine whether to allow a user access to a resource or to the system.

Based on the original user′s request, the resource manager formulates a request to SAF and passes it resource class, resource name, user on whose behalf the request is made, and level of access required. Depending on the type of request, SAF either responds directly or passes the request on to RACF, if it is present. In either case, the resource manager sends the user a response, after it considers SAF′s response.

Neither SAF nor RACF are responsible for enforcing the decision made by SAF. SAF simply returns the decision to the caller (resource manager). The resource manager that requested the resource access checked must enforce that decision made by RACF using SAF.

Access to RACF database may not be required if the response can be formulated from profiles that are in the storage.

Figure 66 shows an overview of the SAF function.

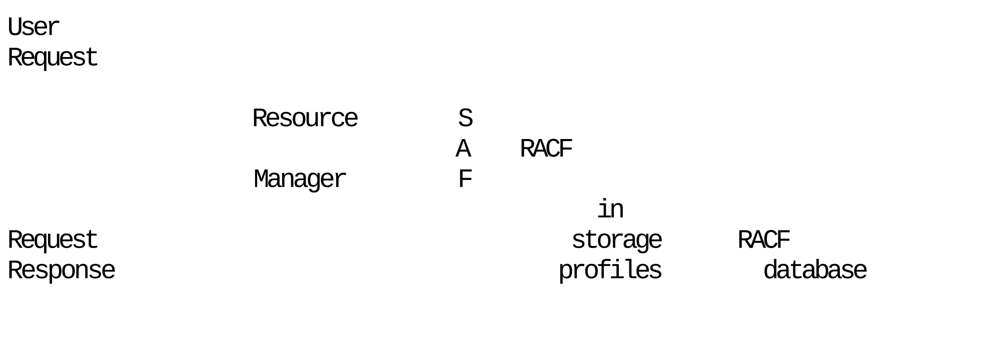


Figure 66. System Authorization Facility Function

For example, JES accesses RACF services through the SAF. Through SAF, JES passes security information about jobs and resources to RACF, which evaluates the security information and returns the result of that evaluation to JES. JES then enforces the results of the security check by, for example, permitting or denying access to a data set or allowing a job to execute. If RACF is inactive or not installed, SAF returns default security information to JES. JES may or may not perform alternative security processing depending on how your system is tailored.

Appendix A. Starter System JCL

This appendix contains the JCL to install the RACF starter system described in this manual.

A.1 Define a Primary Restructured Database

```
//RACRDSP  JOB   (ACCT#),'PGMRNAME',
//         CLASS=A,MSGCLASS=X,MSGLEVEL=(1,1),
//         USER=,GROUP=,PASSWORD=,
//         NOTIFY=,
//         TIME=30
//*
//* LIB: CBIPO   - IPO1.INSTLIB(RACRDSP)
//* GDE: CBIPO MVS INSTALLATION
//* DOC: THIS JOB ALLOCATES, CATALOGS, AND INITIALIZES A
//*      PRIMARY RACF RESTRUCTURED DATA BASE
//*      ON THE SYS627 VOLUME.
//*
//INIT1   EXEC PGM=IRRMIN00,REGION=512K,
//             PARM='NEW'
//STEPLIB  DD  DSN=IPO1SYS.SYS1.LINKLIB,DISP=SHR
//SYSPRINT DD  SYSOUT=*
//SYSTEMP  DD  DSN=IPO1SYS.SYS1.MODGEN(IRRTEMP1),DISP=SHR
//SYSRACF  DD  DSN=IPO1SYS.SYS1.RACF,                        /*R1*/
//             DISP=(NEW,CATLG,DELETE),
//             UNIT=3380,VOL=SER=SYS627,
//             SPACE=(4096,(900),,CONTIG),
//             DCB=(RECFM=F,BLKSIZE=4096,DSORG=PS)
/*
//CATG2   EXEC PGM=IDCAMS,REGION=512K,
//             COND=(0,LT,INIT1)
//SYSPRINT DD  SYSOUT=*
//SYSIN    DD  *
   ALTER -
     IPO1SYS.SYS1.RACF                                        /*R1*/-
     NEWNAME(SYS1.RACF)                                       /*R1*/-
     CATALOG(CATALOG.MVSICFM.VMVSCAT/PWUPDATE)
   IF LASTCC = 0 THEN -
   DEFINE ALIAS( -
     NAME(IPO1SYS.SYS1.RACF)                                  /*R1*/-
     RELATE(SYS1.RACF) )                                      /*R1*/-
     CATALOG(CATALOG.MVSICFM.VMVSCAT/PWUPDATE)
/*
```

A.2 Define a Backup Restructured Database

```
//RACRDSB  JOB   (ACCT#),′PGMRNAME′,
//         CLASS=A,MSGCLASS=X,MSGLEVEL=(1,1),
//         USER=,GROUP=,PASSWORD=,
//         NOTIFY=,
//         TIME=30
//*
//* LIB: CBIPO   - IPO1.INSTLIB(RACRDSB)
//* GDE: CBIPO MVS INSTALLATION
//* DOC: THIS JOB ALLOCATES, CATALOGS, AND INITIALIZES A
//*      BACKUP RACF RESTRUCTURED DATA BASE
//*      (SYS1.RACF.BKUP1)                                     /*R2*/
//*      ON VOLUME MVSDL1
//*
//INIT1   EXEC PGM=IRRMIN00,REGION=512K,
//             PARM=′NEW′
//STEPLIB  DD  DSN=IPO1SYS.SYS1.LINKLIB,DISP=SHR
//SYSPRINT DD  SYSOUT=*
//SYSTEMP  DD  DSN=IPO1SYS.SYS1.MODGEN(IRRTEMP1),DISP=SHR
//SYSRACF  DD  DSN=IPO1SYS.SYS1.RACF.BKUP1,                    /*R2*/
//             DISP=(NEW,CATLG,DELETE),
//             UNIT=3380,VOL=SER=MVSDL1,
//             SPACE=(4096,(900),,CONTIG),
//             DCB=(RECFM=F,BLKSIZE=4096,DSORG=PS)
/*
//CATG2   EXEC PGM=IDCAMS,REGION=512K,
//             COND=(0,LT,INIT1)
//SYSPRINT DD  SYSOUT=*
//SYSIN    DD  *
  ALTER -
     IPO1SYS.SYS1.RACF.BKUP1                                   /*R2*/ -
     NEWNAME(SYS1.RACF.BKUP1)                                  /*R2*/ -
     CATALOG(CATALOG.MVSICFM.VMVSCAT/PWUPDATE)
  IF LASTCC = 0 THEN -
     DEFINE ALIAS( -
     NAME(IPO1SYS.SYS1.RACF.BKUP1)                             /*R2*/ -
     RELATE(SYS1.RACF.BKUP1) )                                 /*R2*/ -
     CATALOG(CATALOG.MVSICFM.VMVSCAT/PWUPDATE)
/*
```

A.3 Define a Primary Non-restructured Database

```
//RACFINT  JOB   (ACCT#),'PGMRNAME',
//         CLASS=A,MSGCLASS=X,MSGLEVEL=(1,1),
//         USER=,GROUP=,PASSWORD=,
//         NOTIFY=,
//         TIME=30
//*
//* LIB: CBIPO   - IPO1.INSTLIB(RACFINT)
//* GDE: CBIPO MVS INSTALLATION
//* DOC: THIS JOB ALLOCATES, CATALOGS, AND INITIALIZES A
//*      PRIMARY RACF NON-RESTRUCTURED DATA BASE
//*      ON THE SYS627 VOLUME.
//*
//INIT1   EXEC PGM=ICHMIN00,REGION=512K,
//             PARM='NEW'
//STEPLIB  DD  DSN=IPO1SYS.SYS1.LINKLIB,DISP=SHR
//SYSPRINT DD  SYSOUT=*
//SYSTEMP  DD  DSN=IPO1SYS.SYS1.MODGEN(ICHTEMP0),DISP=SHR
//SYSRACF  DD  DSN=IPO1SYS.SYS1.RACF,                            /*R1*/
//             DISP=(NEW,CATLG,DELETE),
//             UNIT=3380,VOL=SER=SYS627,
//             SPACE=(1024,(900),,CONTIG),
//             DCB=(RECFM=F,BLKSIZE=1024,DSORG=PS)
/*
//CATG2   EXEC PGM=IDCAMS,REGION=512K,
//             COND=(0,LT,INIT1)
//SYSPRINT DD  SYSOUT=*
//SYSIN    DD  *
   ALTER -
     IPO1SYS.SYS1.RACF                                           /*R1*/-
     NEWNAME(SYS1.RACF)                                          /*R1*/-
     CATALOG(CATALOG.MVSICFM.VMVSCAT/PWUPDATE)
   IF LASTCC = 0 THEN -
   DEFINE ALIAS( -
     NAME(IPO1SYS.SYS1.RACF)                                     /*R1*/-
     RELATE(SYS1.RACF) )                                         /*R1*/-
     CATALOG(CATALOG.MVSICFM.VMVSCAT/PWUPDATE)
/*
```

A.4 Define a Backup Non-restructured Database

```
//RACINBK  JOB   (ACCT#),′PGMRNAME′,
//         CLASS=A,MSGCLASS=X,MSGLEVEL=(1,1),
//         USER=,GROUP=,PASSWORD=,
//         NOTIFY=,
//         TIME=30
//*
//* LIB: CBIPO   - IPO1.INSTLIB(RACINBK)
//* GDE: CBIPO MVS INSTALLATION
//* DOC: THIS JOB ALLOCATES, CATALOGS, AND INITIALIZES A
//*      BACKUP RACF NON-RESTRUCTURED DATA BASE
//*      (SYS1.RACF.BKUP1)                                     /*R2*/
//*      ON VOLUME MVSDL1
//*
//INIT1   EXEC PGM=ICHMIN00,REGION=512K,
//             PARM=′NEW′
//STEPLIB  DD  DSN=IPO1SYS.SYS1.LINKLIB,DISP=SHR
//SYSPRINT DD  SYSOUT=*
//SYSTEMP  DD  DSN=IPO1SYS.SYS1.MODGEN(ICHTEMP0),DISP=SHR
//SYSRACF  DD  DSN=IPO1SYS.SYS1.RACF.BKUP1,                    /*R2*/
//             DISP=(NEW,CATLG,DELETE),
//             UNIT=3380,VOL=SER=MVSDL1,
//             SPACE=(1024,(900),,CONTIG),
//             DCB=(RECFM=F,BLKSIZE=1024,DSORG=PS)
/*
//CATG2   EXEC PGM=IDCAMS,REGION=512K,
//             COND=(0,LT,INIT1)
//SYSPRINT DD  SYSOUT=*
//SYSIN    DD  *
  ALTER -
     IPO1SYS.SYS1.RACF.BKUP1                                   /*R2*/ -
     NEWNAME(SYS1.RACF.BKUP1)                                  /*R2*/ -
     CATALOG(CATALOG.MVSICFM.VMVSCAT/PWUPDATE)
  IF LASTCC = 0 THEN -
     DEFINE ALIAS( -
     NAME(IPO1SYS.SYS1.RACF.BKUP1)                             /*R2*/ -
     RELATE(SYS1.RACF.BKUP1) )                                 /*R2*/ -
     CATALOG(CATALOG.MVSICFM.VMVSCAT/PWUPDATE)
/*
```

A.5 Copy the Primary Non-restructured Database

```
//RACFCPR  JOB   (ACCT#),'PGMRNAME',
//         CLASS=A,MSGCLASS=X,MSGLEVEL=(1,1),
//         USER=,GROUP=,PASSWORD=,
//         NOTIFY=,
//         TIME=30
//*
//* LIB: CBIPO   - IPO1.INSTLIB(RACFCPR)
//* GDE: CBIPO MVS INSTALLATION
//* DOC: THIS JOB COPIES THE PRIMARY RACF DATA BASE
//*      (SYS1.RACF)                                           /*R1*/
//*      ON THE DRIVING SYSTEM TO THE PRIMARY RACF DATA BASE
//*      IPO1SYS.SYS1.RACF                                     /*R1*/
//*      ON THE TARGET SYSTEM.
//*      ANY DISCREPENCIES IN THE INTERNAL ORGANIZATION OF
//*      THE DATA SETS ARE NOTED BY THE INDEX AND MAP CONTROL
//*      STATEMENTS.
//*
//*      IPO1SYS.SYS1.RACF                                     /*R1*/
//*      IS ON SYS627 ON THE TARGET SYSTEM.
//*
//*
//COPYST1 EXEC PGM=ICHUT200,REGION=512K
//STEPLIB  DD  DSN=IPO1SYS.SYS1.LINKLIB,DISP=SHR
//SYSRACF  DD  DSN=SYS1.RACF,                                  /*R1*/
//             DISP=SHR
//SYSUT1   DD  DSN=IPO1SYS.SYS1.RACF,                          /*R1*/
//             DISP=SHR
//SYSUT2   DD  SYSOUT=*
//SYSPRINT DD  SYSOUT=*
//SYSIN    DD  *
 INDEX
 MAP
 END
/*
```

A.6 Copy the Backup Non-restructured Database

```
//RACCPBK  JOB   (ACCT#),′PGMRNAME′,
//         CLASS=A,MSGCLASS=X,MSGLEVEL=(1,1),
//         USER=,GROUP=,PASSWORD=,
//         NOTIFY=,
//         TIME=30
//*
//* LIB: CBIPO   - IPO1.INSTLIB(RACCPBK)
//* GDE: CBIPO MVS INSTALLATION
//* DOC: THIS JOB COPIES THE BACKUP RACF DATA BASE
//*      (SYS1.RACF.BKUP1)                                             /*R2*/
//*      ON THE DRIVING SYSTEM
//*      TO THE BACKUP RACF DATA BASE
//*      IPO1SYS.SYS1.RACF.BKUP1                                       /*R2*/
//*      ON THE TARGET SYSTEM.
//*      ANY DISCREPENCIES IN THE INTERNAL ORGANIZATION OF
//*      THE DATA SETS ARE NOTED BY THE INDEX AND MAP CONTROL
//*      STATEMENTS.
//*
//*      IPO1SYS.SYS1.RACF.BKUP1                                       /*R2*/
//*      IS ON MVSDL1 OF THE TARGET SYSTEM.
//*
//*
//COPYST1 EXEC PGM=ICHUT200,REGION=512K
//STEPLIB  DD  DSN=IPO1SYS.SYS1.LINKLIB,DISP=SHR
//SYSRACF  DD  DSN=SYS1.RACF.BKUP1,                                    /*R2*/
//             DISP=SHR
//SYSUT1   DD  DSN=IPO1SYS.SYS1.RACF.BKUP1,                            /*R2*/
//             DISP=SHR
//SYSUT2   DD  SYSOUT=*
//SYSPRINT DD  SYSOUT=*
//SYSIN    DD  *
 INDEX
 MAP
 END
/*
```

A.7 Upgrade Primary Non-restructured Database Templates

```
//RACUPGNP JOB   (ACCT#),′PGMRNAME′,
//         CLASS=A,MSGCLASS=X,MSGLEVEL=(1,1),
//         USER=,GROUP=,PASSWORD=,
//         NOTIFY=,
//         TIME=30
//*
//* LIB: CBIPO   - IPO1.INSTLIB(RACUPGNP)
//* GDE: CBIPO MVS INSTALLATION
//* DOC: THIS JOB UPGRADES THE PRIMARY NON-RESTRUCTURED RACF
//*      DATA BASE TEMPLATES FROM A PREVIOUS RACF RELEASE
//*      TO THE NEWER RACF RELEASE ON THE SYS627 VOLUME.
//*
//*
//*      THE PRIMARY NON-RESTRUCTURED RACF DATA BASE,
//*      IPO1SYS.SYS1.RACF,                                       /*R1*/
//*      IS ON THE SYS627 VOLUME.
//*
//*
//UPGR1   EXEC PGM=ICHMIN00,REGION=512K,
//             PARM=′UPDATE′
//STEPLIB  DD  DSN=IPO1SYS.SYS1.LINKLIB,DISP=SHR
//SYSPRINT DD  SYSOUT=*
//SYSTEMP  DD  DSN=IPO1SYS.SYS1.MODGEN(ICHTEMP0),DISP=SHR
//SYSRACF  DD  DSN=IPO1SYS.SYS1.RACF,                            /*R1*/
//             DISP=OLD
/*
```

A.8 Upgrade Backup Non-restructured Database Templates

```
//RACUPGNB JOB   (ACCT#),′PGMRNAME′,
//         CLASS=A,MSGCLASS=X,MSGLEVEL=(1,1),
//         USER=,GROUP=,PASSWORD=,
//         NOTIFY=,
//         TIME=30
//*
//* LIB: CBIPO   - IPO1.INSTLIB(RACUPGNB)
//* GDE: CBIPO MVS INSTALLATION
//* DOC: THIS JOB UPGRADES THE BACKUP NON-RESTRUCTURED
//*      RACF DATA BASE TEMPLATES FROM A PREVIOUS
//*      RELEASE OF RACF TO THE NEWER RACF RELEASE.
//*
//*      THE BACKUP NON-RESTRUCTURED RACF DATA BASE,
//*      IPO1SYS.SYS1.RACF.BKUP1,                                /*R2*/
//*      IS ON THE MVSDL1 VOLUME.
//*
//*
//UPGR1   EXEC PGM=ICHMIN00,REGION=512K,
//             PARM=′UPDATE′
//STEPLIB  DD  DSN=IPO1SYS.SYS1.LINKLIB,DISP=SHR
//SYSPRINT DD  SYSOUT=*
//SYSTEMP  DD  DSN=IPO1SYS.SYS1.MODGEN(ICHTEMP0),DISP=SHR
//SYSRACF  DD  DSN=IPO1SYS.SYS1.RACF.BKUP1,                      /*R2*/
//             DISP=OLD
/*
```

A.9 Copy Primary Non-restructured Database to Backup

```
//RACCOPY  JOB   (ACCT#),'PGMRNAME',
//         CLASS=A,MSGCLASS=X,MSGLEVEL=(1,1),
//         USER=,GROUP=,PASSWORD=,
//         NOTIFY=,
//         TIME=30
//*
//* LIB: CBIPO   - IPO1.INSTLIB(RACCOPY)
//* GDE: CBIPO MVS INSTALLATION
//* DOC: THIS JOB COPIES THE PRIMARY
//*      NON-RESTRUCTURED RACF DATA BASE,
//*      IPO1SYS.SYS1.RACF                                     /*R1*/
//*      TO THE BACKUP NON-RESTRUCTURED
//*      RACF DATA BASE,
//*      IPO1SYS.SYS1.RACF.BKUP1                               /*R2*/
//*
//*      IPO1SYS.SYS1.RACF                                     /*R1*/
//*      IS ON THE SYS627 VOLUME.
//*
//*      IPO1SYS.SYS1.RACF.BKUP1                               /*R2*/
//*      IS ON THE MVSDL1 VOLUME.
//*
//*      ANY INCONSISTENCIES IN THE INTERNAL
//*      ORGANIZATION OF THE DATA BASE ARE LISTED
//*      BY THE INDEX AND MAP CONTROL STATEMENTS.
//*
//* NOTE: WHEN YOU SPECIFY DUPLEX WITH NO STATISTICS FOR
//*      THE BACKUP RACF DATA BASE, YOU SHOULD PERIODICALLY
//*      COPY THE PRIMARY DATA BASE TO THE BACKUP DATA BASE
//*      TO ENSURE THAT THE STATISTICS ARE UPDATED.
//*
//COPYST1 EXEC PGM=ICHUT200,REGION=512K
//STEPLIB  DD  DSN=IPO1SYS.SYS1.LINKLIB,DISP=SHR
//SYSRACF  DD  DSN=IPO1SYS.SYS1.RACF,                          /*R1*/
//             DISP=SHR
//SYSUT1   DD  DSN=IPO1SYS.SYS1.RACF.BKUP1,                    /*R2*/
//             DISP=SHR
//SYSUT2   DD  SYSOUT=*
//SYSPRINT DD  SYSOUT=*
//SYSIN    DD  *
 INDEX
 MAP
 END
/*
```

A.10 Copy Primary Restructured Database to Backup

```
//RACRCPY  JOB   (ACCT#),'PGMRNAME',
//         CLASS=A,MSGCLASS=X,MSGLEVEL=(1,1),
//         USER=,GROUP=,PASSWORD=,
//         NOTIFY=,
//         TIME=30
//*
//* LIB: CBIPO   - IPO1.INSTLIB(RACRCPY)
//* GDE: CBIPO MVS INSTALLATION
//* DOC: THIS JOB COPIES THE PRIMARY
//*      RESTRUCTURED RACF DATA BASE,
//*      IPO1SYS.SYS1.RACF                                     /*R1*/
//*      TO THE BACKUP RESTRUCTURED
//*      RACF DATA BASE,
//*      IPO1SYS.SYS1.RACF.BKUP1                               /*R2*/
//*
//*      IPO1SYS.SYS1.RACF                                     /*R1*/
//*      IS ON THE SYS627 VOLUME.
//*
//*      IPO1SYS.SYS1.RACF.BKUP1                               /*R2*/
//*      IS ON THE MVSDL1 VOLUME.
//*
//*      ANY INCONSISTENCIES IN THE INTERNAL
//*      ORGANIZATION OF THE DATA BASE ARE LISTED
//*      BY THE INDEX AND MAP CONTROL STATEMENTS.
//*
//* NOTE: WHEN YOU SPECIFY DUPLEX WITH NO STATISTICS FOR
//*      THE BACKUP RACF DATA BASE, YOU SHOULD PERIODICALLY
//*      COPY THE PRIMARY DATA BASE TO THE BACKUP DATA BASE
//*      TO ENSURE THAT THE STATISTICS ARE UPDATED.
//*
//COPYST1 EXEC PGM=IRRUT200,REGION=512K
//STEPLIB  DD  DSN=IPO1SYS.SYS1.LINKLIB,DISP=SHR
//SYSRACF  DD  DSN=IPO1SYS.SYS1.RACF,                          /*R1*/
//             DISP=SHR
//SYSUT1   DD  DSN=IPO1SYS.SYS1.RACF.BKUP1,                    /*R2*/
//             DISP=SHR
//SYSUT2   DD  SYSOUT=*
//SYSPRINT DD  SYSOUT=*
//SYSIN    DD  *
 INDEX
 MAP
 END
/*
```

A.11 Install RACF Database Name Table

```
//RACDSN   JOB   (ACCT#),′PGMRNAME′,
//         CLASS=A,MSGCLASS=X,MSGLEVEL=(1,1),
//         USER=,GROUP=,PASSWORD=,
//         NOTIFY=,
//         TIME=30
//*
//* LIB: CBIPO   - IPO1.INSTLIB(RACDSN)
//* GDE: CBIPO MVS INSTALLATION
//* DOC: THIS JOB CREATES AN SMP/E USERMOD TO INSTALL
//*      THE RACF DATA BASE NAME TABLE (ICHRDSNT).
//*
//*      ASSEM1: THE RACF DATA BASE NAMES AND DATA BASE OPTIONS
//*              ARE ASSEMBLED INTO AN SMP/E USERMOD, IPOSDS0.
//*              THE OBJECT IS PLACED INTO THE DATA SET &&LOADSET.
//*     RECAPP2: THE SMP/E USERMOD CREATED IN STEP1 IS RECEIVED
//*              AND APPLIED IN THE TARGET LIBRARIES.
//*
//JOBLIB   DD  DSN=IPO1SYS.IPO1.LINKLIB,DISP=SHR
//*
//ASSEM1  EXEC PGM=IEV90,REGION=768K,
//             PARM=′DECK,NOOBJECT′
//SYSLIB   DD  DSN=IPO1SYS.SYS1.AGENLIB,DISP=SHR
//         DD  DSN=IPO1SYS.SYS1.AMODGEN,DISP=SHR
//SYSUT1   DD  UNIT=SYSALLDA,SPACE=(CYL,(20,5))
//SYSUT2   DD  UNIT=SYSALLDA,SPACE=(CYL,(10,1))
//SYSUT3   DD  UNIT=SYSALLDA,SPACE=(CYL,(2,1))
//SYSPRINT DD  SYSOUT=*
//SYSPUNCH DD  DSN=&&LOADSET,DISP=(,PASS),
//             SPACE=(CYL,(5,1)),UNIT=SYSALLDA
//SYSIN    DD  *
         PUNCH ′++USERMOD (IPOSDS0).′
         PUNCH ′++ VER (Z038) FMID(HRF1902).′
         PUNCH ′++ MOD (ICHRDSNT) DISTLIB(AOSBN) LMOD(ICHRDSNT).′
ICHRDSNT CSECT  ,
ICHRDSNT AMODE  31
ICHRDSNT RMODE  24
***********************************************************************
*                                                                     *
*   MODULE NAME = ICHRDSNT                                            *
*                                                                     *
*   DESCRIPTIVE NAME = RACF DATA BASE NAME TABLE.                     *
*                                                                     *
*   FUNCTION = THIS TABLE CONTAINS THE NAMES FOR THE RACF DATA BASES. *
*              -- THE NAMES ARE IN PAIRS, THE FIRST BEING THE         *
*                 PRIMARY DATA BASE AND THE SECOND BEING THE BACKUP   *
*                 DATA BASE.                                          *
*                                                                     *
*   THE FORMAT OF THE FLAG FIELD IS AS FOLLOWS:                       *
*                                                                     *
*   BIT SETTING        MEANING                                        *
*    00.. ....         NO UPDATES TO BACKUP DATA BASE                 *
*    10.. ....         UPDATE BACKUP DATA BASE EXCEPT STATISTICS      *
*    11.. ....         UPDATE BACKUP DATA BASE INCLUDING STATISTICS   *
*    .... ...1         USE THE RESIDENT DATA BLOCK OPTION, IF BIT     *
*                      7 IS ZERO, ONLY INDEX BLOCKS WILL BE KEPT IN   *
*                      STORAGE. OTHERWISE INDEX, BAM, AND PROFILE     *
*                      BLOCKS WILL BE KEPT IN STORAGE.                 *
*                                                                      *
*   RECOMENDED NUMBER OF RESIDENT BLOCKS:                              *
*                                                                      *
*     MVS/XA     = 255                                                 *
*     MVS/ESA    = 255                                                 *
*                                                                      *
************************************************************************
*
         DC   AL1(1)                 NUMBER OF ENTRIES
*
*                                    PRIMARY DATA BASE NAME
         DC   CL44′SYS1.RACF′                                    /*R1*/
*                                    BACKUP DATA BASE NAME
         DC   CL44′SYS1.RACF.BKUP1′                              /*R2*/
         DC   AL1(255)               NUMBER OF RESIDENT BLOCKS
         DC   B′10000001′            DUPLEX UPDATES
*                                    RESIDENT DATA BLOCKS
*
         END
/*
//*
//RECAPP2 EXEC PGM=GIMSMP,REGION=4096K,COND=(0,NE)
//STEPLIB  DD  DSN=IPO1SYS.IPO1.LINKLIB,DISP=SHR
//SMPCSI   DD  DSN=TARGSMP.MVSSMPE.GLOBAL.CSI,DISP=SHR
//MVSTZN   DD  DSN=TARGRES.MVSTZN.CSI,DISP=SHR
//SMPPTS   DD  DSN=TARGSMP.MVSSMPE.SMPPTS,DISP=SHR
//SMPLOG   DD  DSN=IPO1SYS.SYS1.SMPTLOG,DISP=MOD
//SMPOUT   DD  SYSOUT=*
//SMPRPT   DD  SYSOUT=*
//SYSPRINT DD  SYSOUT=*
//SMPWRK1  DD  UNIT=SYSALLDA,SPACE=(6160,(3,3,1)),DISP=(,DELETE),
//             DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//SMPWRK2  DD  UNIT=SYSALLDA,SPACE=(6160,(3,3,1)),DISP=(,DELETE),
//             DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//SMPWRK3  DD  UNIT=SYSALLDA,SPACE=(3120,(5,5,1)),DISP=(,DELETE),
//             DCB=(RECFM=FB,LRECL=80,BLKSIZE=3120)
//SYSUT1   DD  UNIT=SYSALLDA,SPACE=(6160,(3,3,1))
//SYSUT2   DD  UNIT=SYSALLDA,SPACE=(6160,(3,3,1))
//SYSUT3   DD  UNIT=SYSALLDA,SPACE=(6160,(3,3,1))
//SYSUT4   DD  UNIT=SYSALLDA,SPACE=(6160,(3,3,1))
//SMPHOLD  DD  DUMMY
//SMPCNTL  DD  *
  SET BDY(MVSTZN) .
  UCLIN .
  REP LMOD(ICHRDSNT) SYSLIB(LINKLIB) .
  ENDUCL .
  SET BDY(GLOBAL) .
  RECEIVE S(IPOSDS0) SYSMODS .
  SET BDY(MVSTZN) .
  APPLY S(IPOSDS0) .
/*
//SMPPTFIN DD  DSN=&&LOADSET,DISP=(OLD,DELETE)
```

A.12 DES Modification

```
//RACDES   JOB   (ACCT#),'PGMRNAME',
//         CLASS=A,MSGCLASS=X,MSGLEVEL=(1,1),
//         USER=,GROUP=,PASSWORD=,NOTIFY=,TIME=30
//*
//* LIB: CBIPO   - IPO1.INSTLIB(RACDES)
//* GDE: CBIPO MVS INSTALLATION
//* DOC: THIS JOB USES THE SMP/E UCLIN FUNCTION TO CHANGE
//*      THE OWNER OF MODULE ICHDEX01 FROM THE RACF FMID TO
//*      DUMMY FMID IPOSDX0.  AFTER EXECUTING THIS JOB,
//*      ICHDEX01 WILL NOT BE RE-APPLIED WHEN AN MVS SYSGEN
//*      IS RUN OR WHEN A NEW RELEASE OF RACF IS INSTALLED.
//*
//*      DELETE1: DELETE MODULE ICHDEX01
//*      CHANGE2: USE THE UCLIN FUNCTION TO CHANGE THE OWNER
//*               OF ICHDEX01 TO IPOSFMD.
//*
//DELETE1 EXEC PGM=IEHPROGM,REGION=512K
//SYSPRINT DD  SYSOUT=*
//DD1      DD  VOL=SER=MVSRS1,UNIT=3380,
//             DISP=OLD
//SYSIN    DD  *
 SCRATCH       VOL=3380=MVSRS1,DSNAME=SYS1.LPALIB,MEMBER=ICHDEX01
/*

//CHANGE2 EXEC PGM=GIMSMP,REGION=4096K,COND=(0,NE)
//STEPLIB  DD  DSN=IPO1SYS.SYS1.LINKLIB,DISP=SHR
//SMPCSI   DD  DSN=TARGSMP.MVSSMPE.GLOBAL.CSI,DISP=SHR
//MVSTZN   DD  DSN=TARGRES.MVSTZN.CSI,DISP=SHR
//MVSDZN   DD  DSN=TARGDLB.MVSDZN.CSI,DISP=SHR
//SMPPTS   DD  DSN=TARGSMP.MVSSMPE.SMPPTS,DISP=SHR
//SMPLOG   DD  DSN=IPO1SYS.SYS1.SMPTLOG,DISP=MOD
//SMPOUT   DD  SYSOUT=*
//SMPRPT   DD  SYSOUT=*
//SYSPRINT DD  SYSOUT=*
//SMPWRK1  DD  UNIT=SYSALLDA,SPACE=(3120,(5,5,1)),DISP=(,DELETE),
//             DCB=(RECFM=FB,LRECL=80,BLKSIZE=3120)
//SMPWRK2  DD  UNIT=SYSALLDA,SPACE=(3120,(5,5,1)),DISP=(,DELETE),
//             DCB=(RECFM=FB,LRECL=80,BLKSIZE=3120)
//SMPWRK3  DD  UNIT=SYSALLDA,SPACE=(3120,(5,5,1)),DISP=(,DELETE),
//             DCB=(RECFM=FB,LRECL=80,BLKSIZE=3120)
//SYSUT1   DD  UNIT=SYSALLDA,SPACE=(3120,(5,5,1))
//SYSUT2   DD  UNIT=SYSALLDA,SPACE=(3120,(5,5,1))
//SYSUT3   DD  UNIT=SYSALLDA,SPACE=(3120,(5,5,1))
//SYSUT4   DD  UNIT=SYSALLDA,SPACE=(3120,(5,5,1))
//SMPHOLD  DD  DUMMY
//SMPPTFIN DD  DUMMY
//SMPCNTL  DD  *
  SET BDY(MVSTZN) .
   UCLIN .
   REP MOD(ICHDEX01) FMID(IPOSDX0) RMID(IPOSDX0) .
   ENDUCL .
  SET BDY(MVSDZN) .
   UCLIN .
   REP MOD(ICHDEX01) FMID(IPOSDX0) RMID(IPOSDX0) .
   ENDUCL .
/*
```

A.13 Skeleton Group Structure

```
//RACGRP   JOB  (ACCT#),
//         ′PGMRNAME′,                                          /*OPER*/
//         CLASS=A,                                             /*JCLAS*/
//         MSGLEVEL=(1,1),
//         MSGCLASS=X,
//         NOTIFY=,
//         TIME=30,
//         USER=,GROUP=,PASSWORD=                               /*RACF*/
//*                                                             /*JCTRL*/
//*
//* LIB: CBIPO   - IPO1.JCLLIB(RACGRP)
//* GDE: CBIPO SECURITY GUIDE
//* DOC: THIS SAMPLE JCL CREATES AN INITIAL GROUP STRUCTURE.
//*      WITH THIS JOB, YOU CAN DEFINE GROUP NAMES TO
//*      CREATE THE FIRST LEVEL OF A GROUP STRUCTURE
//*      AND THE SYSTEM PROGRAMMER′S ORGANIZATION.
//*
//* NOTE: YOU MUST HAVE THE RACF SPECIAL ATTRIBUTE
//*      TO EXECUTE THIS JOB.
//*
//CREGRP1 EXEC PGM=IKJEFT01,DYNAMNBR=20,REGION=512K
//SYSPRINT DD  DUMMY
//SYSTSPRT DD  SYSOUT=*
//* -------------------------------------------------------------
//* DEFINE THE SYSTEM GROUP STRUCTURE
//* -------------------------------------------------------------
//SYSTSIN  DD  *
ADDGROUP SYSMGT SUPGROUP(SYS1) OWNER(SYS1)
  ADDGROUP SWMGT SUPGROUP(SYSMGT) OWNER(SYSMGT)
    ADDGROUP SYSSW SUPGROUP(SWMGT) OWNER(SWMGT)
    ADDGROUP NETSW SUPGROUP(SWMGT) OWNER(SWMGT)
    ADDGROUP DBSW  SUPGROUP(SWMGT) OWNER(SWMGT)
  ADDGROUP STCGRP SUPGROUP(SYSMGT) OWNER(SYSMGT)
  ADDGROUP STGMGT SUPGROUP(SYSMGT) OWNER(SYSMGT)
    ADDGROUP TAPEMGT SUPGROUP(STGMGT) OWNER(STGMGT)
    ADDGROUP DASDMGT SUPGROUP(STGMGT) OWNER(STGMGT)
ADDGROUP APPLS SUPGROUP(SYS1) OWNER(SYS1)
ADDGROUP USERS SUPGROUP(SYS1) OWNER(SYS1)
/*
```

A.14 SETROPTS

```
//RACSETR  JOB   (ACCT#),
//         'PGMRNAME',                                          /*OPER*/
//         CLASS=A,                                            /*JCLAS*/
//         MSGLEVEL=(1,1),
//         MSGCLASS=X,
//         NOTIFY=,
//         TIME=30,
//         USER=,GROUP=,PASSWORD=                               /*RACF*/
//*                                                            /*JCTRL*/
//*
//* LIB: CBIPO   - IPO1.JCLLIB(RACSETR)
//* GDE: CBIPO SECURITY GUIDE
//* DOC: THIS JOB SETS RACF OPTIONS USING THE SETROPTS
//*      COMMAND.  WITH THIS COMMAND, YOU CAN SET SYSTEM
//*      WIDE RACF OPTIONS RELATED TO RESOURCE PROTECTION.
//*
//*      SOPT1: THE DEFAULT RACF OPTIONS AND THE RECOMMENDED
//*             RACF OPTIONS ARE SET USING SETROPTS.  THIS
//*             CAN BE USED IN THE FUTURE TO RESET RACF
//*             OPTIONS TO THE DESIRED VALUES.
//*
//* NOTE: YOU MUST HAVE THE SPECIAL ATTRIBUTE AND
//*      THE AUDITOR ATTRIBUTE TO EXECUTE THE JOB.
//*
//SOPT1   EXEC PGM=IKJEFT01,DYNAMNBR=20,REGION=512K
//SYSTSPRT DD  SYSOUT=*
//SYSTSIN  DD  *
SETROPTS NOCLASSACT(*)        /***       INITIAL DEFAULT       ***/
SETROPTS TERMINAL(READ)       /* UACC FOR TERMINAL               */
SETROPTS STATISTICS(*)        /***       INITIAL DEFAULT       ***/
SETROPTS INITSTATS            /***       INITIAL DEFAULT       ***/
SETROPTS NOSECLEVELAUDIT      /***       INITIAL DEFAULT       ***/
SETROPTS AUDIT(*)             /* LOG MODIFICATIONS TO PROFILES   */
SETROPTS GENERIC(*)           /* ACTIVATE GENERIC PROFILE CHECK  */
SETROPTS GENCMD(*)            /* ACTIVATE GENERIC PROFILE COMMAND*/
SETROPTS NOEGN                /***       INITIAL DEFAULT       ***/
SETROPTS GLOBAL(*)            /* ACTIVATE GLOBAL ACCESS CHECKING */
SETROPTS NOADSP      /* DEACTIVATE AUTOMATIC DATA SET PROTECTION*/
SETROPTS NOPREFIX             /***       INITIAL DEFAULT       ***/
SETROPTS SAUDIT               /***       INITIAL DEFAULT       ***/
SETROPTS CMDVIOL              /***       INITIAL DEFAULT       ***/
SETROPTS NOINACTIVE           /***       INITIAL DEFAULT       ***/
SETROPTS GRPLIST              /* ACTIVATE LIST-OF-GROUP CHECKING */
SETROPTS NOMODEL              /***       INITIAL DEFAULT       ***/

SETROPTS PASSWORD(HISTORY(32)) /* NUMBER OF PREVIOUS PASSWORDS  */
SETROPTS PASSWORD(REVOKE(5))   /* NUMBER OF INCORRECT PASSWORDS */
SETROPTS PASSWORD(INTERVAL(60)) /* PASSWORD CHANGE INTERVAL (DAYS)*/
SETROPTS PASSWORD(RULE1(LENGTH(6:8) ALPHANUM(1:8))) /*PASSWORD RULE*/
SETROPTS PASSWORD(RULE2(LENGTH(6:8) NUMERIC(1:8))) /*PASSWORD RULE*/
SETROPTS PASSWORD(RULE3(LENGTH(6:8) ALPHA(1:8))) /*PASSWORD RULE*/
SETROPTS PASSWORD(WARNING(8)) /* ISSUE WARNING MESSAGE           */
SETROPTS NOREALDSN            /***       INITIAL DEFAULT       ***/
SETROPTS JES(NOBATCHALLRACF)  /***       INITIAL DEFAULT       ***/
SETROPTS JES(NOXBMALLRACF)    /***       INITIAL DEFAULT       ***/
SETROPTS JES(EARLYVERIFY)     /* INVOKE ICHRTX00 EXIT            */
SETROPTS NOPROTECTALL         /***       INITIAL DEFAULT       ***/
SETROPTS OPERAUDIT           /* LOG ACTIONS OF OPERATIONS USERS */
SETROPTS WHEN(PROGRAM)       /* ACTIVATE PROGRAM CONTROL        */
SETROPTS ERASE               /* ALLOW ERASE-ON-SCRATCH          */
SETROPTS NOTAPEDSN           /***       INITIAL DEFAULT       ***/
SETROPTS RVARYPW(SWITCH(YES)) /***       INITIAL DEFAULT      ***/
SETROPTS RVARYPW(STATUS(YES)) /***       INITIAL DEFAULT      ***/
SETROPTS LIST                 /* LIST CURRENT RACF OPTIONS      */
/*
```

A.15 Global Access Checking Table

```
//RACFGAC  JOB   (ACCT#),
//         'PGMRNAME',                                        /*OPER*/
//         CLASS=A,                                          /*JCLAS*/
//         MSGLEVEL=(1,1),
//         MSGCLASS=X,
//         NOTIFY=,
//         TIME=30,
//         USER=,GROUP=,PASSWORD=                             /*RACF*/
//*                                                          /*JCTRL*/
//*
//* LIB: CBIPO   - IPO1.JCLLIB(RACFGAC)
//* GDE: CBIPO SECURITY GUIDE
//* DOC: THIS JOB CREATES A SAMPLE GLOBAL ACCESS CHECKING
//*      (GAC) TABLE.  RACF PROTECTED RESOURCES THAT ARE
//*      ACCESSED BY MANY PEOPLE SHOULD BE SPECIFIED IN
//*      THIS TABLE.  ACCESS CHECKING IS DONE IN STORAGE
//*      WITHOUT I/O TO THE RACF DATA BASE.
//*
//*      IN THIS SAMPLE:
//*
//*      (1) IF THE DATA SET BEGINS WITH THE RACF USERID OF
//*          THE USER WHO IS ACCESSING THE DATA SET, ALTER
//*          AUTHORITY OR LESS WILL BE ALLOWED.
//*
//*      (2) ANY USER CAN UPDATE SYS1.BRODCAST.
//*
//*      (3) ANY USER CAN READ SYS1.COBLIB, SYS1.PROCLIB,
//*          SYS1.HELP, SYS1.MACLIB, AND ALL ISPF/PDF LIBRARIES.
//*
//CREGAC1 EXEC PGM=IKJEFT01,DYNAMNBR=20,REGION=512K
//SYSPRINT DD  DUMMY
//SYSTSPRT DD  SYSOUT=*
//SYSTSIN  DD  *
  RDEFINE GLOBAL DATASET ADDMEM('&RACUID.*'/ALTER)
    RALTER GLOBAL DATASET ADDMEM('SYS1.BRODCAST'/UPDATE)
    RALTER GLOBAL DATASET ADDMEM('SYS1.COBLIB'/READ)
    RALTER GLOBAL DATASET ADDMEM('SYS1.MACLIB'/READ)
    RALTER GLOBAL DATASET ADDMEM('SYS1.PROCLIB'/READ)
    RALTER GLOBAL DATASET ADDMEM('SYS1.HELP'/READ)
    RALTER GLOBAL DATASET ADDMEM('ISP.*'/READ)
    RALTER GLOBAL DATASET ADDMEM('ISR.*'/READ)
  RLIST GLOBAL *
 /*************************************************************/
 /* IN ORDER TO ACTIVATE THESE CHANGES TO THE GAC TABLE,       */
 /* THE FOLLOWING COMMAND MUST BE ISSUED.                      */
 /*************************************************************/
  SETROPTS GLOBAL(DATASET)
```

```
/*
```

A.16 Started Procedures Table

```
//RACSTC    JOB   (ACCT#),
//          'PGMRNAME',                                        /*OPER*/
//          CLASS=A,                                          /*JCLAS*/
//          MSGLEVEL=(1,1),
//          MSGCLASS=X,
//          NOTIFY=,
//          TIME=30,
//          USER=,GROUP=,PASSWORD=                             /*RACF*/
//*                                                         /*JCTRL*/
//*
//* LIB: CBIPO   - IPO1.JCLLIB(RACSTC)
//* GDE: CBIPO SECURITY GUIDE
//* DOC: THIS JOB CREATES A SAMPLE RACF STARTED PROCEDURES
//*      NAME TABLE.  WITH THIS TABLE, YOU CAN ASSIGN A RACF
//*      USERID AND GROUP NAME TO A STARTED PROCEDURE.
//*
//* NOTE: YOU MUST HAVE THE RACF SPECIAL ATTRIBUTE TO EXECUTE
//*       THE SECOND STEP.
//*
//*    CREATE1: CREATES THE RACF USERIDS FOR THE STARTED
//*             PROCEDURES.  THE RACF GROUPS ASSOCIATED WITH
//*             THESE USERIDS HAVE BEEN CREATED IN THE JOB
//*             RACGRP THAT SHOULD ALREADY HAVE BEEN EXECUTED.
//*
//*    ASSEM2:  THE RACF STARTED PROCEDURE NAME TABLE (ICHRIN03)
//*             IS ASSEMBLED AND THE OBJECT CODE IS PLACED
//*             IN THE DATA SET &&LOADSET.
//*
//*    INSTL3:  THE OBJECT CODE IN &&LOADSET IS INSTALLED AS A
//*             USERMOD WITH SMP/E.
//*
//CREATE1 EXEC PGM=IKJEFT01,DYNAMNBR=20,REGION=512K
//SYSPRINT DD  DUMMY
//SYSTSPRT DD  SYSOUT=*
//SYSTSIN  DD  *
ADDUSER  STCJES OWNER(STCGRP) DFLTGRP(STCGRP) UACC(NONE) NOADSP
ADDSD   'STCJES.*'  OWNER(STCGRP) UACC(NONE) AUDIT(SUCCESS(UPDATE))
ADDUSER  STCDPT OWNER(STCGRP) DFLTGRP(STCGRP) UACC(NONE) NOADSP
ADDSD   'STCDPT.*'  OWNER(STCGRP) UACC(NONE) AUDIT(SUCCESS(UPDATE))
PASSWORD USER( STCJES, STCDPT ) INTERVAL(30)
 LISTGRP  STCGRP
 LISTUSER STCJES
 LISTUSER STCDPT
/*
//*
//ASSEM2  EXEC ASMHC,
//             PARM='DECK,NOOBJECT',
//             COND=(0,NE)
//SYSPUNCH DD  DSN=&&LOADSET,UNIT=SYSALLDA,DISP=(,PASS),
//             SPACE=(CYL,(5,1))
//SYSPRINT DD  SYSOUT=*
//SYSIN    DD  *
         PUNCH '++USERMOD (IPOSST0).'
         PUNCH '++ VER (Z038) FMID(HRF1902).'
         PUNCH '++ MOD (ICHRIN03) DISTLIB(AOSBN) LMOD(ICHRIN03).'
************************************************************************
*                                                                      *
*   MODULE NAME = ICHRIN03                                             *
*                                                                      *
*   DESCRIPTIVE NAME = RACF STARTED PROCEDURES NAME TABLE              *
*                                                                      *
*   FUNCTION =  THIS TABLE ASSIGNS USERID AND GROUP NAME TO A          *
*               -- STARTED PROCEDURE.                                  *
*               -- FOR DESCRIPTION SEE SYSTEM PROGRAMMING              *
*                  LIBRARY: RACF AND COMMENTS IN MODULE.               *
*                                                                      *
************************************************************************
ICHRIN03 TITLE 'RACF STARTED PROCEDURES NAME TABLE'
ICHRIN03 CSECT
ICHRIN03 AMODE 31
ICHRIN03 RMODE ANY
         DC    AL2(ENTRIES+X'8000')          NUMBER OF ENTRIES FOLLOWED
         SPACE 3
FIRST    DS    0H
************************************************************************
*   INSTALLATION DEFINED ENTRIES                                       *
************************************************************************
*        DC    C'PROC    USER    GROUP   '    EXAMPLE LINE
         DC    C'JES2    STCJES  STCGRP  '    JES2
         DC    X'40'                          FLAG BYTE - TRUSTED
         DC    X'00000000000000'              RESERVED
         DC    C'JES3    STCJES  STCGRP  '    JES3
         DC    X'40'                          FLAG BYTE - TRUSTED
         DC    X'00000000000000'              RESERVED
         DC    C'IRRDPTABSTCDPT  STCGRP  '    DYNAMIC PARSE TABLE
         DC    X'00'                          FLAG BYTE - NEITHER
         DC    X'00000000000000'              RESERVED
         SPACE 2

* ***************************************************************
*                                                               *
*  THE DEFAULT USERID '=' IS ASSIGNED TO ALL PROCEDURES         *
*  WHICH ARE NOT DEFINED ABOVE.  THIS GENERIC ENTRY MUST        *
*  BE LAST IN THE TABLE.                                        *
*                                                               *
* ***************************************************************
         DC    C'*       =       STCGRP  '    OTHER PROCEDURES
         DC    X'00'                          FLAG BYTE - NEITHER
         DC    X'00000000000000'              RESERVED
**********************************************************************
*   END OF ENTRY LIST                                                *
**********************************************************************
LAST     DS    0H
ENTLEN   EQU   32                             ENTRY LENGTH
TOTLEN   EQU   LAST-FIRST                     TOTAL LENGTH
ENTRIES  EQU   TOTLEN/ENTLEN                  NUMBER OF ENTRIES
         END   ICHRIN03
/*
//*
//INSTL3  EXEC IPOSMPE,COND=(0,NE),
//             CSI='MVSSMPE.GLOBAL.CSI'
//SMPEIN   DD  *
  SET BDY(MVSTZN) .
  UCLIN .
  REP LMOD(ICHRIN03) SYSLIB(LPALIB).
  ENDUCL .
  SET BDY(GLOBAL) .
  RECEIVE S(IPOSST0) SYSMODS .
  SET BDY(MVSTZN) .
  APPLY S(IPOSST0) .
/*
//SMPPTFIN DD  DSN=&&LOADSET,DISP=(OLD,DELETE)
```

A.17 Router Exit

```
//RACRTX0  JOB   (ACCT#),
//         'PGMRNAME',                                       /*OPER*/
//         CLASS=A,                                          /*JCLAS*/
//         MSGLEVEL=(1,1),
//         MSGCLASS=X,
//         NOTIFY=,
//         TIME=30,
//         USER=,GROUP=,PASSWORD=                            /*RACF*/
//*                                                          /*JCTRL*/
//*
//* LIB: CBIPO   - IPO1.JCLLIB(RACRTX0)
//* GDE: CBIPO SECURITY GUIDE
//* DOC: THIS JOB CREATES A USERMOD TO INSTALL THE MVS ROUTER
//*      EXIT (ICHRTX00) IN SMP/E FORMAT.
//*
//*      THIS SAMPLE EXIT ROUTINE IS NOT NEEDED IF THE
//*      INSTALLATION IS USING MVS 3.1.3 AND JES 3.1.3
//*      OR HIGHER
//*
//*
//CREATE1 EXEC ASMHC,
//             PARM='DECK,NOOBJECT'
//SYSLIB   DD  DSN=IPO1SYS.SYS1.MACLIB,DISP=SHR
//         DD  DSN=IPO1SYS.SYS1.AMODGEN,DISP=SHR
//SYSPUNCH DD  DSN=&&LOADSET,DISP=(,PASS),SPACE=(CYL,(5,1)),
//             UNIT=SYSALLDA
//SYSPRINT DD  SYSOUT=*
//*
//SYSIN    DD  *
         PUNCH '++USERMOD (IPOSRT0).'
         PUNCH '++ VER (Z038) FMID(HRF1902).'
         PUNCH '++ MOD (ICHRTX00) DISTLIB(AOSBN) LMOD(ICHRTX00).'
ICHRTX00 TITLE 'MVS Router exit for JOB verification.'
***********************************************************************
***********************************************************************
***                                                                 ***
***      MVS Router (SAF) exit for job verification.                ***
***                                                                 ***
***      NOTE: This sample exit routine is not needed if the        ***
***            installation is using MVS 3.1.3 AND JES 3.1.3    @A1A***
***                                                                 ***
***      This sample exit routine performs the following functions: ***
***                                                                 ***
***      1. Checks if the request is VERIFY (RACINIT) and the       ***
***         'ENVIR' parameter in the RACINIT parameter list is      ***
***         'VERIFY' (X'C0'). in all other cases, the exit returns  ***
***         with return code zero (pass control to RACF Router).    ***
***                                                                 ***
***      2. If a password (or an old/new password combination)      ***
***         has been specified, a RACINIT macro is issued (with     ***
***         the ’ENVIR=CREATE’ parameter to verify the combination  ***
***         of userid and password.                                 ***
***                                                                 ***
***         The user is verified if RACINIT returns with return     ***
***         code 0. In this case, the router exit returns control   ***
***         with return code 200 (changed to 0 by the MVS router).  ***
***         any other return code will cause a return code 208      ***
***         with the RACINIT return and reason codes stored in the  ***
***         SAFPRRET and SAFPRREA return and reason code fields.    ***
***                                                                 ***
***      3. If a userid has been specified without a password,      ***
***         the address of the input parameter list (CACP) of       ***
***         module IEFCMAUT (the issuer of the RACROUTE request)    ***
***         is located by pointing back to IEFCMAUT’s save area     ***
***         via the save area chain and picking up register 1.      ***
***         the parameter list is verified.                         ***
***         The submitting user’s authority to submit a job with    ***
***         the specified userid is verified by:                    ***
***                                                                 ***
***         a) Issuing a RACINIT macro for the submitting user’s    ***
***            userid with the ’ENVIR=CREATE’ and ’PASSCHK=NO’      ***
***            parameters.                                          ***
***         b) Issuing a RACHECK macro for class ’FACILITY’ with    ***
***            ’read’ as required attribute and a profile name      ***
***            of ’$SUBMIT.userid’ (the job’s userid).              ***
***                                                                 ***
***         The user is verified if RACHECK returns with return     ***
***         code 0. In this case, the router exit returns control   ***
***         with return code 200 (changed to 0 by the MVS router).  ***
***         Any other return code will cause a return code 208      ***
***         with SAFPRRET return code X’18’ (failed by installation ***
***         exit) and the RACHECK return code in SAFPRREA.          ***
***         If the CACP parameter list cannot be found, return code ***
***         204 reason code 4 will be used.                         ***
***                                                                 ***
***      4. If the submitting userid is absent and the password on  ***
***         the submitted job is absent, the job will be failed     ***
***         to cause the IEF722I - INVALID PASSWORD GIVEN message.  ***
***                                                                 ***
***********************************************************************
***********************************************************************
         EJECT
         SPACE 2
***********************************************************************
***********************************************************************
***                                                                 ***
***      Registers on entry:                                        ***
***      R1    Pointer to exit parameter list                       ***
***            Offset 0 - ptr. to SAFP parameter list               ***
***            Offset 4 - ptr. to 152-byte work area                ***
***                                                                 ***
***      Exit:                                                      ***
***      R15   0 - Pass control to RACF router                      ***
***          200 - Authorization granted, RACF processing bypassed  ***
***                This is returned for return code 0 from          ***
***                RACINIT or RACHECK, respectively.                ***
***          204 - Submitter's userid could not be located in the   ***
***                CACP parameter list. No exit processing.         ***
***          208 - Authorization denied, RACF processing bypassed   ***
***                Returned for any other return code from          ***
***                RACINIT or RACHECK, respectively.                ***
***                                                                 ***
***      REGISTER USAGE:                                            ***
***      R13   SAVE AREA                                            ***
***      R12   PROGRAM BASE                                         ***
***      R10   ADDR. OF WORK AREA  (RTXWORK)                        ***
***      R9    ADDR. OF EXIT PARAMETER LIST (ICHSAFP)               ***
***      R8    ADDR. OF RACINIT PARAMETER LIST                      ***
***      R6    Saved reason code                                    ***
***      R5    Saved return code                                    ***
***                                                                 ***
***      DEPENDENCIES:                                              ***
***      1. JES support for RACF 1.6 must be installed.             ***
***         JES2: PTF UZ90341      JES3: PTF UZ90352                ***
***         plus corequisite scheduler ptf's                        ***
***      2. Save area chain from module IEFCMAUT over ICHSFR00      ***
***         to ICHRTX00. reg.1 on  entry to IEFCMAUT must be        ***
***         ptr. to addr. of CACP.                                  ***
***      3. The RACINIT macros are using RACF 1.7 keywords of       ***
***         STAT and RELEASE.                                       ***
***      4. The ICHRAC assembler global variable is used to         ***
***         generate 31-bit RACHECK and RACINIT expansions.         ***
***                                                                 ***
***                                                                 ***
***      MVS/XA AMODE and RMODE considerations:                     ***
***        RMODE(24)                                                ***
***        AMODE(ANY)                                               ***
***                                                                 ***
***********************************************************************
***********************************************************************
         EJECT
         SPACE 5
***********************************************************************
***********************************************************************
***                                                                 ***
***      Copyright IBM corporation, 1985                            ***
***                                                                 ***
***      This code has not been submitted to any formal IBM test    ***
***      and is distributed on an "as is" basis without any         ***
***      warranty either express or implied. The implementation     ***
***      of any of the techniques described or used herein is a     ***
***      customer responsibility and depends on the customer's      ***
***      operational environment. While each item may have been     ***
***      reviewed for accuracy in a specific situation and may      ***
***      run in a specific environment, there is no guarantee       ***
***      that the same or similar results will be obtained else-    ***
***      where. Customers attempting to adapt these techniques to   ***
***      their own environments do so at their own risk.            ***
***                                                                 ***
***********************************************************************
***********************************************************************
         SPACE  3
***********************************************************************
***********************************************************************
***                                                                 ***
***      Mapping macros           Macro library                   ***
***      --------------           -------------                   ***
***      ICHSAFP                  SYS1.MACLIB                     ***
***      IEFCMAUP                 SYS1.MODGEN                     ***
***      IHAACEE                  SYS1.MACLIB                     ***
***                                                               ***
***      Load module name         System library                  ***
***      ----------------         --------------                  ***
***      ICHRTX00 (RENT,REFR)     SYS1.LPALIB                     ***
***                                                               ***
***********************************************************************
***********************************************************************
         PRINT OFF
         MACRO
&NAME    REGI  &BIT=YES,&FLOAT=NO
         GBLA  &IEZBITS
***********************************************************************
*                                                                     *
*         Register and bit equates                                    *
*                                                                     *
*         REGI  BIT=NO     Register equates only.                     *
*         REGI  FLOAT=YES  Floating point registers too.              *
*                                                                     *
***********************************************************************

*
R0       EQU   0
R1       EQU   1
R2       EQU   2
R3       EQU   3
R4       EQU   4
R5       EQU   5
R6       EQU   6
R7       EQU   7
R8       EQU   8
R9       EQU   9
R10      EQU   10
R11      EQU   11
R12      EQU   12
R13      EQU   13
R14      EQU   14
R15      EQU   15
         AIF   ('&FLOAT' NE 'YES').BITS
*
F0       EQU   0                   Floating point register 0
F2       EQU   2                   Floating point register 2
F4       EQU   4                   Floating point register 4
F6       EQU   6                   Floating point register 6
.BITS    ANOP
         AIF   ('&BIT' NE 'YES').END
         AIF   (&IEZBITS EQ 0).SETSW
         MEXIT
.SETSW   ANOP
&IEZBITS SETA  1
*
BIT0     EQU   X'80'
BIT1     EQU   X'40'
BIT2     EQU   X'20'
BIT3     EQU   X'10'
BIT4     EQU   X'08'
```

```
BIT5     EQU   X’04’
BIT6     EQU   X’02’
BIT7     EQU   X’01’
.END     ANOP
         MEND
         PRINT ON
ICHRTX00 CSECT
ICHRTX00 AMODE ANY
ICHRTX00 RMODE 24
         GBLB  &ICHRAC
         EJECT
         ICHSAFP
         EJECT
         IEFCMAUP
         EJECT
         IHAACEE
         EJECT
*
***********************************************************************
*                                                                     *
*        Work area for the MVS router exit (152 bytes max.)           *
*        This area is part of the 512-byte workarea passed by the     *
*        issuer of the RACROUTE macro instruction.                    *
*                                                                     *
***********************************************************************
*
RTXWORK  DSECT
RTXACR   DS    CL4                 Acronym (RTXW)
RTXACEE  DS    A                   ACEE address.
RTXJOBN  DS    CL8                 Job name.
* Following values (RTXJUSER, RTXULEN, RTXSUSER, RTXGLEN, RTXSGRP) are
* overlaid when the RACHECK parameter list is built.
RTXJUSER DS    CL8                 Job’s userid.
RTXULEN  DS    C                   Length of submitter’s userid.
RTXSUSER DS    CL8                 Submitter’s userid.
RTXGLEN  DS    C                   Length of submitter’s group name.
RTXSGRP  DS    CL8                 Submitter’s connect group.
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
         ORG   RTXJUSER            Overlay areas no longer used.
RTXPROF  EQU   *                   Start of profile name.
RTXPPREF DS    CL(L’PROFPFX)       Prefix for profile.
RTXPUSER DS    CL(L’ACEEUSRI)      Userid in profile.
* Value of 39 from class descriptor table definition for FACILITY.
RTXRMNDR DS    CL(39-L’PROFPFX-L’ACEEUSRI)
*
         DS    0F
ROUTEPFX EQU   *                   Start of RACROUTE prefix area.
         RACROUTE REQUEST=VERIFY,WORKA=*-*,                            *
               ACEE=*-*,MF=L       RACINIT parameter list.
*
         ORG   ROUTEPFX            Overlay RACINIT area.
         DS    0F
         RACROUTE REQUEST=AUTH,WORKA=*-*,                              *
               ATTR=READ,ACEE=*-*,MF=L RACHECK parameter list.
*
         ORG
RTXLNG   EQU   *-RTXWORK           Length of work area
         EJECT
         REGI  BIT=YES
         EJECT
ICHRTX00 CSECT
         SAVE  (14,12),,ICHRTX00.RACF19.&SYSDATE
         LR    R12,R15
         USING ICHRTX00,R12        Program base
         USING SAFP,R9             Base for SAFP parameter list
         L     R9,0(R1)            Get addr. of SAF parameter list
*
***********************************************************************
*                                                                     *
*        Return to MVS router if not REQUEST=VERIFY,ENVIR=VERIFY      *
*                                                                     *
***********************************************************************
*
         SR    R15,R15             Return code 0
         LA    R2,SAFPVER          Request=verify request type
         CH    R2,SAFPREQT         Check request type
         BNE   GOBACK              -not request=verify, return
         L     R8,SAFPRACP         Offset to RACINIT parameter list
         AR    R8,R9               Addr. of RACINIT parameter list.
*
         TM    3(R8),BIT0+BIT1     Is it envir=verify ?
         BNO   GOBACK              -no, back to caller
*
***********************************************************************
*                                                                     *
*        Establish workarea base, check if password specified        *
*                                                                     *
***********************************************************************
*
         SR    R2,R2               Load with zero.
         ST    R2,SAFPRRET         Initialize return code
         ST    R2,SAFPRREA         Initialize reason code
         USING RTXWORK,R10         Base for work area
         L     R10,4(R1)           Get addr. of workarea
         MVC   RTXACR,=C'RTXW'     Acronym to work area.
         ICM   R2,15,8(R8)         Get ptr. to password
*                                   from RACINIT.
         BZ    NOPASWD             -zero, no password specified
         CLI   0(R2),0             Is password length zero?
         BE    NOPASWD             -yes, no password specified
         EJECT
*
***********************************************************************
*                                                                     *
*        Issue RACINIT for userid/password on JOB card.               *
*                                                                     *
***********************************************************************
*
         MVC   ROUTEPFX(RACINC1L),RACINCR1 Move skeleton.
         LA    R1,ROUTEPFX
         L     R2,4(,R8)           Address of userid from RACINIT.
         L     R3,8(,R8)           Address of password from RACINIT.
         L     R4,20(,R8)          Address of group from RACINIT.
         L     R5,24(,R8)          Address of newpaswd from RACINIT.
```

```
  SETB  1                 31-bit expansion.
         RACINIT MF=(E,(1)),ACEE=RTXACEE,                              *
               USERID=(R2),                                            *
               GROUP=(R4),                                             *
               PASSWRD=(R3),                                           *
               NEWPASS=(R5)
  SETB  0                 24-bit expansion.
         RACINIT MF=(E,(1))
         LTR   R15,R15              Check return code.
         BZ    RACINOK              -Zero, branch.
         LA    R5,8                 Return code 8 from RACROUTE.
         ST    R15,SAFPRRET         Store return code.
         ST    R0,SAFPRREA          Store reason code.
         B     RACINDEL             Delete ACEE
*
RACINOK  DS    0H
         SR    R5,R5                Return code 0.
*
***********************************************************************
*                                                                     *
*        Issue RACINIT ENVIR=DELETE if RACINIT was successful.        *
*                                                                     *
***********************************************************************
*
RACINDEL DS    0H
         ICM   R2,15,RTXACEE        Addr. of ACEE.
         BZ    RETCODES             -Zero, don't delete ACEE.
         MVC   ROUTEPFX(RACINDL),RACINDE Move skeleton.
         LA    R1,ROUTEPFX
  SETB  1                 31-bit expansion.
         RACINIT ACEE=RTXACEE,SMC=NO,MF=(E,(1))  ENVIR=DELETE
  SETB  0                 24-bit expansion.
*        R1 set up to L-form area.
         RACINIT MF=(E,(1))         ENVIR=DELETE
         LTR   R15,R15              Successful?
         BZ    RETCODES             -Yes, branch.
DUMPER   ABEND 100,REASON=(15)      Cause user abend.
RETCODES DS    0H
         LA    R15,200(R5)          Return code 200 or 208.
         B     GOBACK               Back to caller.
         EJECT
*
***********************************************************************
*                                                                     *
*        Locate submitter's userid in CACP (IEFCMAUT param. list)     *
*                                                                     *
***********************************************************************
*

NOPASWD  DS    0H
         L     R2,4(R13)            Get MVS router's save area
         L     R3,4(R2)             Get addr. of IEFCMAUT's save area
         L     R1,24(R3)            Get reg.1 from save area
         L     R4,0(R1)             Pick up CACP addr. from reg.1
         USING CACP,R4              Base for CACP parameter list
         CLC   CACID,=C'CACP'       Is acronym ok?
         BNE   NOCACP               -no, wrong address of CACP
         MVC   RTXJUSER,BLANKS      Initialize with blanks
         L     R2,4(R8)             Ptr. to job's userid (RACINIT)
         SR    R1,R1
         IC    R1,0(R2)            Get userid length
         BCTR  R1,0                Subtract 1 for execute
         EX    R1,JUSERMV          Executed move to RTXJUSR
         L     R1,CACJUSER         Get addr. of job userid
         CLC   RTXJUSER,0(R1)      Same userid in CACP?
         BNE   NOCACP              -no, wrong address of CACP
*
         L     R1,CACSUSER         Get addr. of submitter userid
         MVC   RTXSUSER,0(R1)      Copy submitter's userid
         LA    R1,L'RTXSUSER
         LA    R3,RTXSUSER-1       Point before userid
BLKLP1   DS    0H
         LA    R2,0(R1,R3)         Point to char. in userid
         CLI   0(R2),C' '          Is it a blank ?
         BH    FNDUSR              -no, branch
         BCT   R1,BLKLP1           Bump to next character
         LA    R5,8                Fake in Invalid Password.
         ST    R5,SAFPRRET         Store return code.
         ST    R5,SAFPRREA         Store reason code.
         B     RETCODES            Quit.
FNDUSR   DS    0H
         STC   R1,RTXULEN          Store length of userid
*
         L     R1,CACSGRP          Get addr. of submitter group name
         MVC   RTXSGRP,0(R1)       Copy submitter's connect group
         LA    R1,L'RTXSGRP
         LA    R3,RTXSGRP-1        Point before group name
BLKLP2   DS    0H
         LA    R2,0(R1,R3)         Point to char. in group name
         CLI   0(R2),C' '          Is it a blank ?
         BH    FNDGRP              -no, branch
         BCT   R1,BLKLP2           Bump to next character
FNDGRP   DS    0H
         STC   R1,RTXGLEN          Store length of group name
*
         L     R1,CACJOBNM         Get addr. of job name
         MVC   RTXJOBN,0(R1)       Copy job name
         DROP  R4                  Release base for CACP
*
************************************************************************
*                                                                      *
*        Issue RACINIT (No pswd check) for submitting userid           *
*                                                                      *
************************************************************************
*
         MVC   ROUTEPFX(RACINC2L),RACINCR2 Move skeleton.
         LA    R1,ROUTEPFX
&ICHRAC  SETB  1                   31-bit expansion.
         RACINIT USERID=RTXULEN,GROUP=RTXGLEN,                         X
               JOBNAME=RTXJOBN,ACEE=RTXACEE,                           X
               MF=(E,(1))          RACINIT ENVIR=CREATE
&ICHRAC  SETB  0                   24-bit expansion.
*        R1 set up to L-form area.
         RACINIT MF=(E,(1))        Racinit ENVIR=CREATE
         LTR   R15,R15             Check return code
         BZ    RACINCOK            -zero, branch
         LA    R5,8                Return code 8 from RACROUTE
         ST    R15,SAFPRRET        Store return code
         ST    R0,SAFPRREA         Store reason code
         B     RACINDEL            Delete ACEE and back to caller
*
JUSERMV  MVC   RTXJUSER(0),1(R2)   Executed move of job's userid
*
***********************************************************************
*                                                                     *
*        Issue RACHECK for class 'FACILITY' for job's userid          *
*                                                                     *
***********************************************************************
*
RACINCOK DS    0H
         MVC   RTXPUSER,RTXJUSER    Move job userid.
         MVC   RTXPPREF,PROFPFX     Prefix portion of name.
         MVC   RTXRMNDR,PROFSFX     Add in blank padding.
         MVC   ROUTEPFX(RACHECKL),RACHECK Move skeleton.
         LA    R1,ROUTEPFX
         L     R2,RTXACEE           Temporary ACEE address.
&ICHRAC  SETB  1                   31-bit expansion.
         RACHECK ENTITY=RTXPROF,                                       X
               CLASS=FACILITY,                                         X
               APPL=RTXJOBN,ACEE=(R2),MF=(E,(1))
&ICHRAC  SETB  0                   24-bit expansion.
*        R1 set up to L-form area for the RACHECK expansion.
         RACHECK MF=(E,(1))        Now issue SVC.
         LTR   R0,R15              Check return code.
         BZ    CHECKOK             -Zero, branch.
*

*        RACHECK return codes can be 4 (Profile not found) or
*        8 (Access not allowed). The IEF722I messages (USER PROFILE
*        NOT FOUND and WRONG PASSWORD GIVEN) would be misleading,
*        therefore the return code is changed to 18 (Failed by inst.
*        exit) and the RACHECK return code is stored in the reason
*        code field.
*
         LA    R15,X'18'           Change to 'Failed by inst  exit'
         LA    R5,8                Tell router to fail job.
         ST    R15,SAFPRRET        Store return code.
         ST    R0,SAFPRREA         Store reason code.
         B     RACINDEL            Delete ACEE and return.
CHECKOK  DS    0H
         SR    R5,R5               Return code 0 (Submit allowed).
         B     RACINDEL            Delete ACEE and return.
*
***********************************************************************
*                                                                     *
*        CACP parameter list not found. Return code 204               *
*                                                                     *
***********************************************************************
*
NOCACP   DS    0H
         LA    R0,4                Reason code 4 (router exit)
         ST    R0,SAFPRREA         Store reason code
         ST    R0,SAFPRRET         Store retcode 4 in SAP parmlist
         LA    R15,204             Return code 204 from router exit
*
***********************************************************************
*                                                                     *
*        Return to caller, return code in register 15                 *
*        Reg.0 not restored, reason code in register 0                *
*                                                                     *
***********************************************************************
*
GOBACK   DS    0H
         L     R14,12(R13)         Restore return address
         RETURN (1,12),RC=(15)     Back to caller
*
***********************************************************************
*                                                                     *
*        CONSTANTS                                                    *
*                                                                     *
***********************************************************************
*
BLANKS   DC    CL8' '
*
FACILITY DC    AL1(8),CL8'FACILITY'
PROFPFX  DC    C'$SUBMIT.'
PROFSFX  DC    CL(39-L'PROFPFX-L'ACEEUSRI)' '
         ORG
         SPACE 3
         LTORG
         EJECT
*
***********************************************************************
*                                                                     *
*        LIST-FORM MACRO INSTRUCTION PROTOTYPES.                      *
*                                                                     *
***********************************************************************
*
RACINCR1 RACROUTE WORKA=*-*,REQUEST=VERIFY,                            *
               ENVIR=CREATE,ACEE=0,SMC=NO,PASSCHK=YES,MF=L,            *
               STAT=NO,RELEASE=1.9
RACINC1L EQU   *-RACINCR1          Length of parameter list.
*
RACINCR2 RACROUTE WORKA=*-*,REQUEST=VERIFY,                            *
               ENVIR=CREATE,ACEE=0,SMC=NO,PASSCHK=NO,MF=L,             *
               STAT=NO,RELEASE=1.9
RACINC2L EQU   *-RACINCR2          Length of parameter list.
*
RACINDE  RACROUTE WORKA=*-*,REQUEST=VERIFY,                            *
               ENVIR=DELETE,ACEE=0,SMC=NO,MF=L
RACINDL  EQU   *-RACINDE           Length of parameter list.
*
RACHECK  RACROUTE WORKA=*-*,REQUEST=AUTH,                              *
               MF=L,ATTR=READ,ACEE=*-*,CLASS=*-*
RACHECKL EQU   *-RACHECK           Length of parameter list.
         END
/*
//*
//SMPE2    EXEC IPOSMPE,COND=(0,NE),
//             CSI='MVSSMPE.GLOBAL.CSI'
//SMPEIN   DD  *
  SET BDY(MVSTZN) .
  UCLIN .
  ADD LMOD(ICHRTX00) SYSLIB(LPALIB) .
  ENDUCL .
  SET BDY(GLOBAL) .
  RECEIVE S(IPOSRT0) SYSMODS .
  SET BDY(MVSTZN) .
```

```
   APPLY S(IPOSRT0) .
/*
//SMPPTFIN DD  DSN=&&LOADSET,DISP=(OLD,DELETE)
```

A.18 RACFRW Predefined Reports

```
//RACFRW   JOB   (ACCT#),'PGMRNAME',
//         CLASS=A,                                        /*JCLAS*/
//         MSGLEVEL=(1,1),
//         MSGCLASS=X,NOTIFY=,TIME=30,
//         USER=,GROUP=,PASSWORD=                           /*RACF*/
//*                                                        /*JCTRL*/
//*
//* LIB: CBIPO   - IPO1.JCLLIB(RACWPW)
//*                          (RACWWAR)
//*                          (RACWCMD)
//*                          (RACWAV)
//*                          (RACWOPE)
//*                          (RACWSPE)
//*                          (RACWALL)
//*
//* GDE: CBIPO SECURITY GUIDE
//* DOC: THIS JOB PRODUCES PRE-DEFINED RACFRW (RACF REPORT WRITER)
//*      REPORTS.  USE THE APPROPRIATE SYSTSIN AND RACFRW CONTROL
//*      STATEMENTS IN THIS EXAMPLE TO PRODUCE THE DESIRED RACF REPORTS.
//*
//* NOTE: YOU NEED THE FOLLOWING DDNAMES FOR ALL REPORTS.  ON THE
//*       STATEMENT INDICATED, EDIT AND CHANGE VOL=SER=TAPE01 TO YOUR
//*       VOLSER.
//*
//*      RSMFIN   = THE SMF RECORDS FOR INPUT TO RACFRW
//*      SYSPRINT = THE RACF REPORT
//*      SORTIN   = SORT INPUT FROM RACFRW
//*      SORTLIB  = THE LIBRARY THAT CONTAINS THE SORT/MERGE
//*                  LOAD MODULES
//*      SORTDDNM = SORT MESSAGES
//*      SORTWKXX = SORT WORK DATA SETS
//*      SYSTSIN  = RACFRW CONTROL STATEMENTS
//*
//VIOLAT1 EXEC PGM=IKJEFT01,DYNAMNBR=20,REGION=512K
//SYSPRINT DD  SYSOUT=*
//RSMFIN   DD  DSN=IPOBAK.SMFDUMP(0),DISP=SHR,
//             UNIT=3400-6,VOL=SER=TAPE01,                 <===CHG1
//             LABEL=(,SL)
//SYSTSPRT DD  SYSOUT=*
//SORTDDNM DD  SYSOUT=*
//SORTIN   DD  UNIT=SYSALLDA,SPACE=(CYL,(5,1))
//SORTWK01 DD  UNIT=SYSALLDA,SPACE=(CYL,(5,1))
//SORTWK02 DD  UNIT=SYSALLDA,SPACE=(CYL,(5,1))
//SORTWK03 DD  UNIT=SYSALLDA,SPACE=(CYL,(5,1))
//SORTWK04 DD  UNIT=SYSALLDA,SPACE=(CYL,(5,1))
//SORTWK05 DD  UNIT=SYSALLDA,SPACE=(CYL,(5,1))
//SORTLIB  DD  DSN=IPO1SYS.SYS1.SORTLIB,                   <--VERIFY/C
//             DISP=SHR
```

```
//********************************************************/
//*                                                      */
//*     RACWPW - PASSWORD VIOLATIONS REPORT              */
//*     ------   --------------------------              */
//*                                                      */
//*    THESE STATEMENTS PRODUCE A REPORT OF PASSWORD     */
//*    VIOLATIONS AT LOGON TIME.                         */
//*                                                      */
//********************************************************/
//*
//SYSTSIN  DD  *
 RACFRW  TITLE (’PASSWORD VIOLATION REPORT’)
   SELECT PROCESS
    EVENT LOGON EVQUAL(1)
 LIST SORT(USER)
 END
/*
//********************************************************/
//*                                                      */
//*   RACWWAR - MONITORING RACF IN WARNING MODE REPORT   */
//*   -------   --------------------------------------   */
//*                                                      */
//*  THESE STATEMENTS PRODUCE A REPORT OF ACCESS GRANTED */
//*  ONLY BECAUSE THE WARNING MODE IS ACTIVE.            */
//*                                                      */
//********************************************************/
//*
//SYSTSIN  DD  *
 RACFRW  TITLE (’MONITORING ACCESS ATTEMPTS IN WARNING MODE’)
   SELECT PROCESS  WARNING
 LIST SORT(DATE USER EVENT NAME CLASS)
 END
/*
//********************************************************/
//*                                                      */
//*     RACWCMD - RACF COMMANDS REPORT                   */
//*     -------   --------------------                   */
//*                                                      */
//*   THESE STATEMENTS PRODUCE A LIST OF ALL THE RACF    */
//*   COMMANDS ISSUED.  THE REPORT IS SORTED BY USER.    */
//*                                                      */
//********************************************************/
//SYSTSIN  DD  *
 RACFRW  TITLE (’MONITORING THE USE OF RACF COMMANDS’)
  SELECT PROCESS
   EVENT ALLCOMMAND
  SUMMARY COMMAND BY (USER)
 END
/*

//********************************************************/
//*                                                      */
//*     RACWAV - ACCESS VIOLATIONS REPORT                */
//*     ------   ------------------------                */
//*                                                      */
//*  THESE STATEMENTS PRODUCE A REPORT OF ACCESS FAILURES */
//*  TO ALL RACF PROTECTED RESOURCES AND VIOLATIONS AT   */
//*  LOGON TIME.                                         */
//*  A SUMMARY REPORT BY RESOURCE AND BY USER WILL       */
//*  ALSO BE PROVIDED.                                   */
//*                                                        */
//*********************************************************/
//SYSTSIN  DD  *
 RACFRW  TITLE (′MONITORING ACCESS VIOLATIONS′)
   SELECT PROCESS VIOLATIONS
    EVENT ACCESS EVQUAL(1)
    EVENT LOGON
   LIST SORT(USER  CLASS)
   SUMMARY RESOURCE BY(USER)
   SUMMARY USER BY(RESOURCE)
 END
/*
//*********************************************************/
//*                                                        */
//*     RACWOPE - OPERATIONS USERS REPORT                  */
//*     -------   ----------------------                   */
//*                                                        */
//*  THESE STATEMENTS PRODUCE A REPORT OF THE ACTIVITIES  */
//*  OF OPERATIONS AND GROUP-OPERATIONS USERS.             */
//*                                                        */
//*********************************************************/
//*
//SYSTSIN  DD  *
 RACFRW TITLE(′USERS WITH OPERATIONS OR GROUP-OPERATIONS ATTRIBUTE′ )
   SELECT PROCESS AUTHORITY(OPERATIONS)
    EVENT ALLSVC
    EVENT ALLCOMMAND
 LIST SORT(USER)
 END
/*

//*********************************************************/
//*                                                        */
//*     RACWSPE - SPECIAL USERS REPORT                     */
//*     -------   --------------------                     */
//*                                                        */
//*  THESE STATEMENTS PRODUCE A REPORT OF THE ACTIVITIES  */
//*  OF SPECIAL AND GROUP-SPECIAL USERS.                   */
//*                                                        */
//*********************************************************/
//*
//SYSTSIN  DD  *
 RACFRW  TITLE (′USERS WITH SPECIAL OR GROUP-SPECIAL ATTRIBUTE REPORT′)
   SELECT PROCESS REASON(SPECIAL)
   SELECT PROCESS AUTHORITY(SPECIAL)
     EVENT ALLSVC
     EVENT ALLCOMMAND
 LIST SORT(USER)
 END
/*
//*********************************************************/
//*                                                        */
//*     RACWALL - ALL ACTIVITIES REPORT                    */
//*     -------   --------------------                     */
//*                                                        */
//*  THESE STATEMENTS PRODUCE A REPORT OF ALL RACF SMF     */
//*  RECORDS SORTED BY USER, AND A  GENERAL SUMMARY        */
//*  REPORT SHOWING OVERALL RACF-RELATED SYSTEM ACTIVITY. */
//*  THIS REPORT CAN PRODUCE MORE THAN 10,000 LINES        */
//*                                                        */
```

```
//*********************************************************/
//*
//SYSTSIN  DD  *
 RACFRW  TITLE (’ALL ACTIVITIES REPORT BY USER’) GENSUM
  LIST SORT(USER)
 END
/*
```

A.19 DSMON

```
//RACDSM   JOB   (ACCT#),
//         ’PGMRNAME’,                                         /*OPER*/
//         CLASS=A,                                           /*JCLAS*/
//         MSGLEVEL=(1,1),
//         MSGCLASS=X,
//         NOTIFY=,
//         TIME=30,
//         USER=,GROUP=,PASSWORD=                              /*RACF*/
//*                                                           /*JCTRL*/
//*
//* LIB: CBIPO   - IPO1.JCLLIB(RACDSM)
//* GDE: CBIPO SECURITY GUIDE
//* DOC: THIS SAMPLE JCL PRODUCES DSMON (DATA SECURITY MONITOR)
//*      REPORTS.  DSMON PRODUCES REPORTS ON THE STATUS OF THE
//*      SECURITY ENVIRONMENT AND, IN PARTICULAR, ON THE
//*      STATUS OF THE RACF RESOURCES.  YOU NEED THREE CONTROL
//*      STATEMENTS TO CONTROL DSMON REPORTING:
//*
//*      LINECOUNT = NUMBER OF LINES PER PAGE
//*
//*      FUNCTION  = FUNCTION NAME
//*        SYSTEM  = SYSTEM REPORT
//*        RACGRP  = GROUP TREE REPORT
//*        SYSPPT  = PROGRAM PROPERTIES TABLE REPORT
//*        RACAUT  = RACF AUTHORIZED CALLER TABLE REPORT
//*        RACCDT  = RACF CLASS DESCRIPTOR TABLE REPORT
//*        RACEXT  = RACF EXITS REPORT
//*        RACGAC  = RACF GLOBAL ACCESS TABLE REPORT
//*        RACSPT  = RACF STARTED PROCEDURE TABLE REPORT
//*        RACUSR  = SELECTED USERS ATTRIBUTE REPORT
//*        SYSAPF  = SELECTED DATA SETS REPORT
//*        SYSLNK  = SELECTED DATA SETS REPORT
//*        SYSCAT  = SELECTED DATA SETS REPORT                 .NU..RU.
//*        RACDST  = SELECTED DATA SETS REPORT
//*        SYSSDS  = SELECTED DATA SETS REPORT
//*        USEROPT = SELECTED DATA SETS REPORT(USED
//*                  WITH USRDSN OR RACGRP REPORT)
//*        ALL     = ALL THE REPORTS
//*
//*        USEROPT = USER INPUT TO BE PROCESSED BY THE
//*                  FUNCTION SPECIFIED.  THE VALID
//*                  FUNCTIONS THAT YOU CAN SPECIFY ARE:
//*                     USRDSN  = STATUS OF SELECTED USER DATASET
//*                     RACGRP  = LIST THE GROUP TREE OF A SPECIFIC
//*                               GROUP NAME
//*
```

```
//FUNCT1  EXEC PGM=ICHDSM00,REGION=512K
//SYSPRINT DD  SYSOUT=*
//SYSUT2   DD  SYSOUT=*
//SYSUT1   DD  DSN=IPO1SYS.SYS1.PARMLIB,DISP=SHR
//SYSIN    DD  *
  FUNCTION  SYSTEM
  FUNCTION  RACGRP                      /*  REMOVE THE FUNCTIONS YOU */
  FUNCTION  SYSPPT                      /*  DO NOT WANT TO INCLUDE   */
  FUNCTION  RACAUT
  FUNCTION  RACCDT
  FUNCTION  RACEXT
  FUNCTION  RACGAC
  FUNCTION  RACSPT
  FUNCTION  RACUSR
  FUNCTION  SYSAPF
  FUNCTION  SYSLNK
  FUNCTION  SYSCAT                                         /* .NU..RU. */
  FUNCTION  RACDST
  FUNCTION  SYSSDS
  FUNCTION  ALL                  /* THIS FUNCTION GENERATES ALL THE */
                                 /* REPORTS, IF YOU SELECT THIS,    */
                                 /* REMOVE ALL THE STATEMENTS ABOVE */
  FUNCTION  USRDSN                              /* USEROPT STATEMENTS */
   USEROPT   USRDSN SYS1.LINKLIB                /* CHANGE DS NAME     */
  FUNCTION  RACGRP
   USEROPT   RACGRP SYS1                        /* CHANGE GROUP NAME  */
/*
```

Appendix B. Records Generated by Unload Utility

Table 2. Records Generated by the RACF Database Unload Utility IRRDBU00.

Record type	Description
0100	Group basic data
0101	Group subgroups
0102	Group members
0103	Group installation data
0110	Group DFP data
0200	User basic data
0201	User categories
0202	User classes
0203	User group connections
0204	User installation data
0205	User connect data
0210	User DFP data
0220	User TSO data
0230	User CICS data
0240	User language data
0250	User OPERPARM data
0251	User OPERPARM scope
0260	User WORKATTR data
0400	Data set basic data
0401	Data set categories
0402	Data set conditional access
0403	Data set volumes
0404	Data set access
0405	Data set installation data
0410	Data set DFP data
0500	General resource basic data
0501	General resource tape volumes
0502	General resource categories
0503	General resource members
0504	General resource volumes
0505	General resource access
0506	General resource installation data
0507	General resource conditional access data
0510	General resource session data
0511	General resource session entities
0520	General resource DLF data
0521	General resource DLF job names

Appendix C. RACF Report Writer Output Samples

This appendix contains sample RACF Reports produced from the RACFRW predefined reports.

C.1 Header page

This header is included in all outputs from RACF report writer. It will not be repeated in the examples below, but use to be listed for all reports.

```
 EVENT/QUALIFIER KEY ------
    EVENT  QUALIFIER  MEANING
      1               JOB INITIATION / TSO LOGON/LOGOFF
               0       SUCCESSFUL INITIATION
               1       INVALID PASSWORD
               2       INVALID GROUP
               3       INVALID OIDCARD
               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 SUBMIT JOB
              20       WARNING-INSUFFICIENT SECURITY LABEL AUTHORITY
              21       WARNING-SECURITY LABEL MISSING FROM JOB, USER, OR PROFILE
              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       OIDCARD IS REQUIRED
              30       NETWORK JOB ENTRY - JOB NOT AUTHORIZED
              31       WARNING-UNKNOWN USER FROM TRUSTED NODE PROPAGATED
      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 CATALOGED
              11       DATA SET NOT CATALOGED
              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 LABL
              16       WARNING: NEW NAME NOT PROTECTED BY SECURITY LABL
              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 SECURITY LABE
       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 (INCLUDES PERMFILE, PERMDIR)
      20               RALTER COMMAND (INCLUDES ALTFILE, ALTDIR)
      21               RDEFINE COMMAND (INCLUDES ADDFILE, ADDDIR)
      22               RDELETE COMMAND (INCLUDES DELFILE, DELDIR)
      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 PARTNERLU
                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
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 VMEVENT DESCRIPTION MEANS THAT THE EVENT CONTINUES ON THE NEXT LINE
  .A ’(T)’ APPENDED TO A DATASET IN LIST OF DATASET NAMES AFFECTED BY A SECLABEL CHANGE MEANS THAT THE DATASET IS A TAPE DATASET.
```

C.2 All Activities Report

```
92.086 16:20:59                                    RACF REPORT                                       PAGE   1
                                           ALL ACTIVITIES REPORT BY USER
COMMAND GROUP ENTERED -
  RACFRW FORMAT TITLE(’ALL ACTIVITIES REPORT BY USER’) GENSUM
  LIST SORT(USER)
  END
92.086 16:20:59                            RACF REPORT - GENERAL SUMMARY                             PAGE   4
                                           ALL ACTIVITIES REPORT BY USER
                                                        READ    SELECTED    %-SELECTED
STATUS RECORDS                                            0          0         0 %
PROCESS RECORDS                                         903        903       100 %
TOTAL PROCESS RECORDS FOR DEFINED USERS                 903        903       100 % (OF ALL PROCESS RECORDS)
TOTAL PROCESS RECORDS FOR UNDEFINED USERS                 0          0         0 % (OF ALL PROCESS RECORDS)
                                         --- JOB / LOGON  STATISTICS ---
TOTAL JOB/LOGON/LOGOFF                                  179
TOTAL JOB/LOGON SUCCESSES                                76                   42 % OF TOTAL ATTEMPTS
TOTAL JOB/LOGON VIOLATIONS                                1                    1 % OF TOTAL ATTEMPTS
TOTAL JOB/LOGON ATTEMPTS BY UNDEFINED USERS               0                    0 % OF TOTAL ATTEMPTS
TOTAL JOB/LOGON SUCCESSES BY UNDEFINED USERS              0                    0 % OF TOTAL ATTEMPTS
TOTAL JOB/LOGON VIOLATIONS BY UNDEFINED USERS             0                    0 % OF TOTAL ATTEMPTS
TOTAL JOB/LOGON SUCCESSFUL TERMINATION                  102
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          0          0          0          0
         16-17      17-18      18-19      19-20      20-21      21-22      22-23      23-24
             1          0          0          0          0          0          0          0
```

```
                                                    --- RESOURCE  STATISTICS ---
TOTAL RESOURCE ACCESSES (ALL EVENTS)                              724
TOTAL RESOURCE ACCESS SUCCESSES                                   720                   99 % OF TOTAL ACCESSES
TOTAL RESOURCE ACCESS WARNINGS                                      0                    0 % OF TOTAL ACCESSES
TOTAL RESOURCE ACCESS VIOLATIONS                                    4                    1 % 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                     369                   51 % OF TOTAL ACCESSES
TOTAL RESOURCE ACCESS SUCCESSES USING GENERIC PROFILE             365                   50 % OF TOTAL ACCESSES
TOTAL RESOURCE ACCESS WARNINGS USING GENERIC PROFILE                0                    0 % OF TOTAL ACCESSES
TOTAL RESOURCE ACCESS VIOLATIONS USING GENERIC PROFILE              4                    1 % 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           0           0           0           0
         16-17       17-18       18-19       19-20       20-21       21-22       22-23       23-24
             3           0           0           0           0           0           0           1


92.086 16:20:59                               RACF REPORT - LISTING OF PROCESS RECORDS                                     PAGE   5
                                                    ALL ACTIVITIES REPORT BY USER
                                                        E
                                                        V  Q
                                                        E  U
                     *JOB/USER  *STEP/    --TERMINAL--  N  A
 DATE    TIME   SYSID   NAME     GROUP       ID    LVL  T  L
92.083 17:30:18 XA01   ARCURI    TSO       SCGSJ044   0  7  0  JOBID=(ARCURI 92.083 12:56:58),USERDATA=()
                       M. ARCURI                               AUTH=(NORMAL),REASON=(CLASS)
                                                               SESSION=TSO LOGON,TERMINAL=SCGSJ044
                                                               DATASET=NPM.V1R5M0.FNMPARM,VOLUME=MXASC3
92.083 17:30:22 XA01   ARCURI    TSO       SCGSJ044   0  5  0  JOBID=(ARCURI 92.083 12:56:58),USERDATA=(),OWNER=SYS1
                       M. ARCURI                               AUTH=(NORMAL),REASON=(CLASS)
                                                               SESSION=TSO LOGON,TERMINAL=SCGSJ044
                                                               DATASET=NPM.V1R5M0.FNMPROFS,GENPROF=NPM.*,VOLUME=MXATP1,LEVEL=00
92.083 19:12:48 XA01   ARCURI    TSO       SCGSJ044   0  2  0  JOBID=(ARCURI 92.083 18:12:39),USERDATA=(),OWNER=SYS1
                       M. ARCURI                               AUTH=(NORMAL),REASON=(ENTITY OR FAILSOFT PROCESSING)
                                                               SESSION=TSO LOGON,TERMINAL=SCGSJ044
                                                               DATASET=SYS1.LOCAL.VTAMLIB,GENPROF=SYS1.*,VOLUME=CCSGXX,LEVEL=00,
                                                               INTENT=UPDATE,ALLOWED=UPDATE
92.083 20:16:53 XA01   ARCURI    TSO       SCGSJ044   0  1  8  JOBID=(ARCURI 92.083 18:12:39),USERDATA=()
                                                               AUTH=(NONE),REASON=(NONE)
92.083 22:30:03 XA01   AUDITOR   SYS1                 0  1  0  JOBID=(RACFAUDT 92.083 22:30:01),USERDATA=()
                                                               AUTH=(NONE),REASON=(NONE)
92.083 16:47:08 XA01   DODELL    SYS1      SCGSQ097   0  1  0  JOBID=(DODELL 92.083 16:47:06),USERDATA=()
                                                               AUTH=(NONE),REASON=(NONE)
92.083 16:47:11 XA01   DODELL    SYS1      SCGSQ097   0  2  0  JOBID=(DODELL 92.083 16:47:06),USERDATA=(),OWNER=SYS1
                       DAVE ODELL                              AUTH=(NORMAL),REASON=(ENTITY OR FAILSOFT PROCESSING)
                                                               SESSION=TSO LOGON,TERMINAL=SCGSQ097
                                                               DATASET=SYS1.DDIR,GENPROF=SYS1.*,VOLUME=MXACAT,LEVEL=00,INTENT=
                                                               CONTROL,ALLOWED=ALTER
92.083 16:52:29 XA01   DODELL    SYS1                 0  1  0  JOBID=(DEFPAGE 92.083 16:52:28),USERDATA=()
                                                               AUTH=(NONE),REASON=(NONE)
92.083 16:52:30 XA01   DODELL    SYS1                 0  7  0  JOBID=(DEFPAGE 92.083 16:52:28),USERDATA=()
                       DAVE ODELL                              AUTH=(NORMAL),REASON=(CLASS)
                                                               SESSION=INTERNAL READER BATCH JOB,JESINPUT=INTRDR,EXENODE=WTSCMXA,
                                                               SUBMITTING USER=DODELL,SUBMITTING NODE=WTSCMXA,SUBMITTING GROUP=SYS1
                                                               DATASET=DBP2.DSNDBC.DB01.TS01.I0001.A001,VOLUME=STDB2B
92.083 19:25:58 XA01   HANDEL    TSO       SCGSA064   0  5  0  JOBID=(HANDEL 92.083 13:07:25),USERDATA=(),OWNER=HANDEL
                       HANDEL/CHIAKPO                          AUTH=(NORMAL),REASON=(CLASS)
                                                               SESSION=TSO LOGON,TERMINAL=SCGSA064
                                                               DATASET=HANDEL.SPFLOG1.LIST,GENPROF=HANDEL.*,VOLUME=MXSC05,LEVEL=00
92.083 19:26:01 XA01   HANDEL    TSO       SCGSA064   0  1  8  JOBID=(HANDEL 92.083 13:07:25),USERDATA=()
                                                               AUTH=(NONE),REASON=(NONE)
92.083 16:37:35 XA01   P2010AA   P2010     SCDLD12B   0  2  1  JOBID=(P2010AA 92.083 16:31:24),USERDATA=(),OWNER=GITTINS
                       ADRIAN ALVES                            AUTH=(NORMAL),REASON=(ENTITY OR FAILSOFT PROCESSING)
                                                               SESSION=TSO LOGON,TERMINAL=SCDLD12B
                                                               DATASET=GITTINS.A.A,GENPROF=GITTINS.*,VOLUME=INFOPK,LEVEL=00,INTENT=
                                                               READ,ALLOWED=NONE
92.083 16:37:45 XA01   P2010AA   P2010     SCDLD12B   0  2  1  JOBID=(P2010AA 92.083 16:31:24),USERDATA=(),OWNER=GITTINS
                       ADRIAN ALVES                            AUTH=(NORMAL),REASON=(ENTITY OR FAILSOFT PROCESSING)
                                                               SESSION=TSO LOGON,TERMINAL=SCDLD12B
                                                               DATASET=GITTINS.A.A,GENPROF=GITTINS.*,VOLUME=INFOPK,LEVEL=00,INTENT=
                                                               READ,ALLOWED=NONE
92.083 22:43:50 XA01   ROGERS    P9113                0  1  0  JOBID=(ROGERSI 92.083 22:43:48),USERDATA=()
                                                               AUTH=(NONE),REASON=(NONE)
92.083 16:52:25 XA01   STADE8    TSO       SNR227IB   0  2  0  JOBID=(STADE8 92.083 10:31:19),USERDATA=(),OWNER=STADE8
                       BERTRAND DUFRASNE                       AUTH=(NORMAL),REASON=(ENTITY OR FAILSOFT PROCESSING)
                                                               SESSION=TSO LOGON,TERMINAL=SNR227IB
                                                               DATASET=STADE8.ISPF.ISPPROF,GENPROF=STADE8.*.*,VOLUME=MXATSO,LEVEL=
                                                               00,INTENT=UPDATE,ALLOWED=ALTER
92.083 22:02:18 XA01   STC       SYS1                 0  4  0  JOBID=(DFHSM 92.083 06:57:47),USERDATA=(),OWNER=DODELL
                           (NAME UNKNOWN)                      AUTH=(NORMAL),REASON=(CLASS)
                                                               SESSION=STARTED PROCEDURE,TOKEN STATUS=(
                                                               CREATED BY PRE 1.9 RACF CALL)
                                                               DATASET=HSM.OCDS.BACKUP.V0000505,GENPROF-FROM=HSM.*,GENPROF-TO=HSM.,
                                                               VOLUME=MXATSO,LEVEL=00,NEW RESOURCE NAME=HSM.OCDS.BACKUP.V0000509
```

C.3 Access Violations Report

```
92.086 16:21:07                                             RACF REPORT                                                   PAGE   1
                                                    MONITORING ACCESS VIOLATIONS
COMMAND GROUP ENTERED -
  RACFRW FORMAT TITLE(′MONITORING ACCESS VIOLATIONS′)
  SELECT PROCESS VIOLATIONS
  EV ACCESS EVQUAL(1)
  EV LOGON
  LIST SORT(USER CLASS)
  SUMMARY RESOURCE BY(USER)
  SUMMARY USER BY(RESOURCE)
  END
92.086 16:21:07                               RACF REPORT - LISTING OF PROCESS RECORDS                                    PAGE   4
                                                    MONITORING ACCESS VIOLATIONS
                                                        E
                                                        V  Q
                                                        E  U
                     *JOB/USER  *STEP/    --TERMINAL--  N  A
 DATE    TIME   SYSID   NAME     GROUP       ID    LVL  T  L
92.083 16:37:35 XA01   P2010AA   P2010    SCDLD12B   0  2  1  JOBID=(P2010AA 92.083 16:31:24),USERDATA=(),OWNER=GITTINS
                       ADRIAN ALVES                           AUTH=(NORMAL),REASON=(ENTITY OR FAILSOFT PROCESSING)
                                                              SESSION=TSO LOGON,TERMINAL=SCDLD12B
                                                              DATASET=GITTINS.A.A,GENPROF=GITTINS.*,VOLUME=INFOPK,LEVEL=00,INTENT=
                                                              READ,ALLOWED=NONE
92.083 16:37:45 XA01   P2010AA   P2010    SCDLD12B   0  2  1  JOBID=(P2010AA 92.083 16:31:24),USERDATA=(),OWNER=GITTINS
                       ADRIAN ALVES                           AUTH=(NORMAL),REASON=(ENTITY OR FAILSOFT PROCESSING)
                                                              SESSION=TSO LOGON,TERMINAL=SCDLD12B
                                                              DATASET=GITTINS.A.A,GENPROF=GITTINS.*,VOLUME=INFOPK,LEVEL=00,INTENT=
                                                              READ,ALLOWED=NONE
92.083 16:39:28 XA01   P2010AA   P2010    SCDLD12B   0  2  1  JOBID=(P2010AA 92.083 16:31:24),USERDATA=(),OWNER=GITTINS
                       ADRIAN ALVES                           AUTH=(NORMAL),REASON=(ENTITY OR FAILSOFT PROCESSING)
                                                              SESSION=TSO LOGON,TERMINAL=SCDLD12B
                                                              DATASET=GITTINS.ZZ810320.SCRIPT,GENPROF=GITTINS.*,VOLUME=MXAESE,
                                                              LEVEL=00,INTENT=READ,ALLOWED=NONE
92.083 16:31:02 XA01   P2010AA   P2010    SCDLD12B   0  1 25  JOBID=( 00.000 00:00:00),USERDATA=()
                       ADRIAN ALVES                           AUTH=(NONE),REASON=(RACINIT FAILURE)
                                                              SESSION=TSO LOGON,TERMINAL=SCDLD12B
92.083 23:09:14 XA01   STCSP4    TSO      SJA2002B   0  2  1  JOBID=(STCSP4 92.083 23:04:45),USERDATA=(),OWNER=PATMC
                       P.MCCARTHY                             AUTH=(NORMAL),REASON=(ENTITY OR FAILSOFT PROCESSING)
                                                              SESSION=TSO LOGON,TERMINAL=SJA2002B
                                                              DATASET=PATMC.CSP330.MSL,GENPROF=PATMC.*,VOLUME=STIMS5,LEVEL=00,
                                                              INTENT=UPDATE,ALLOWED=READ

92.086 16:21:07                                RACF REPORT - RESOURCE BY USER SUMMARY                                     PAGE   5
                                                    MONITORING ACCESS VIOLATIONS
      USER/                                                                       ------------ I N T E N T S ------------
      *JOB                                            SUCCESS   WARNING VIOLATION     ALTER   CONTROL    UPDATE      READ     TOTAL
DATASET =+GITTINS.A.A
    P2010AA    ADRIAN ALVES                                 0         0         2         0         0         0         2         2
    ACCUMULATED TOTALS -                                    0         0         2         0         0         0         2         2
    PERCENTAGE OF TOTAL ACCESSES -                        0 %       0 %     100 %       0 %       0 %       0 %     100 %
    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         2         0         0         0         2         2
    PERCENTAGE OF TOTAL ACCESSES -                        0 %       0 %     100 %       0 %       0 %       0 %     100 %
DATASET =+GITTINS.ZZ810320.SCRIPT
    P2010AA    ADRIAN ALVES                                 0         0         1         0         0         0         1         1
    ACCUMULATED TOTALS -                                    0         0         1         0         0         0         1         1
    PERCENTAGE OF TOTAL ACCESSES -                        0 %       0 %     100 %       0 %       0 %       0 %     100 %
    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         1         0         0         0         1         1
    PERCENTAGE OF TOTAL ACCESSES -                        0 %       0 %     100 %       0 %       0 %       0 %     100 %
DATASET =+PATMC.CSP330.MSL
    STCSP4     P.MCCARTHY                                   0         0         1         0         0         1         0         1
    ACCUMULATED TOTALS -                                    0         0         1         0         0         1         0         1
    PERCENTAGE OF TOTAL ACCESSES -                        0 %       0 %     100 %       0 %       0 %     100 %       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         1         0         0         1         0         1
    PERCENTAGE OF TOTAL ACCESSES -                        0 %       0 %     100 %       0 %       0 %     100 %       0 %

92.086 16:21:07                                RACF REPORT - USER BY RESOURCE SUMMARY                                     PAGE   7
                                                    MONITORING ACCESS VIOLATIONS
                                                                                  ------------ I N T E N T S ------------
                  RESOURCE NAME                       SUCCESS   WARNING VIOLATION     ALTER   CONTROL    UPDATE      READ     TOTAL
USER =  P2010AA
  NAME =     ADRIAN ALVES
  CLASS = DATASET
   +GITTINS.A.A                                             0         0         2         0         0         0         2         2
   +GITTINS.ZZ810320.SCRIPT                                 0         0         1         0         0         0         1         1
  ACCUMULATED TOTALS -                                      0         0         3         0         0         0         3         3
  PERCENTAGE OF TOTAL ACCESSES -                          0 %       0 %     100 %       0 %       0 %       0 %     100 %
  GENERIC PROFILE USED
  ACCUMULATED TOTALS -                                      0         0         3         0         0         0         3         3
  PERCENTAGE OF TOTAL ACCESSES -                          0 %       0 %     100 %       0 %       0 %       0 %     100 %
USER =  STCSP4
  NAME =     P.MCCARTHY
  CLASS = DATASET
   +PATMC.CSP330.MSL                                        0         0         1         0         0         1         0         1
  ACCUMULATED TOTALS -                                      0         0         1         0         0         1         0         1
  PERCENTAGE OF TOTAL ACCESSES -                          0 %       0 %     100 %       0 %       0 %     100 %       0 %
  GENERIC PROFILE USED
  ACCUMULATED TOTALS -                                      0         0         1         0         0         1         0         1
  PERCENTAGE OF TOTAL ACCESSES -                          0 %       0 %     100 %       0 %       0 %     100 %       0 %
```

C.4 RACF Commands Report

```
 92.086 16:21:11                                              RACF REPORT                                                   PAGE   1
                                                  MONITORING THE USE OF RACF COMMANDS
 COMMAND GROUP ENTERED -
   RACFRW FORMAT TITLE('MONITORING THE USE OF RACF COMMANDS')
   SELECT PROCESS
   EV ALLCOMMAND
   SUMMARY COMMAND BY(USER)
   END
 92.086 16:21:11                                 RACF REPORT - COMMAND BY USER SUMMARY                                      PAGE   4
                                                  MONITORING THE USE OF RACF COMMANDS
                         QUALIFIER                OCCURRENCES    USER         NAME
 EVENT = 19 - PERMIT/PERMFILE/PERMDIR COMMAND
             0 - NO VIOLATIONS DETECTED
                                                          1      P5104YB    BELLEDAME/WHITE
                 ACCUMULATED TOTALS -                     1
             ACCUMULATED TOTALS -                         1
```

C.5 Special Users Report

```
 92.086 16:21:25                                              RACF REPORT                                                   PAGE   1
                                                  USERS WITH SPECIAL ATTRIBUTE REPORT
 COMMAND GROUP ENTERED -
   RACFRW FORMAT TITLE('USERS WITH SPECIAL ATTRIBUTE REPORT')
   SELECT PROCESS REASON(SPECIAL)
   SELECT PROCESS AUTHORITY(SPECIAL)
   EV ALLSVC
   EV ALLCOMMAND
   LIST SORT(USER)
   END
192.087 10:18:34                               RACF REPORT - LISTING OF PROCESS RECORDS                                     PAGE   4
0                                                 USERS WITH SPECIAL ATTRIBUTE REPORT
0                                                        E
                                                         V  Q
                                                         E  U
                      *JOB/USER  *STEP/    --TERMINAL--  N  A
  DATE    TIME   SYSID   NAME     GROUP       ID    LVL  T  L
092.086 19:58:48 XA01   PATRONE   TSO      SCGSM062   0 19  0  JOBID=(PATRONE 92.086 16:55:57),USERDATA=(),OWNER=DODELL
                        M. PATRONE                             AUTH=(SPECIAL),REASON=(CLASS,SPECIAL/OPERATIONS)
                                                               SESSION=TSO LOGON,TERMINAL=SCGSM062
                                                                 PERMIT P1101.* CLASS(DATASET) ID(VAINI) ACCESS(ALTER)
```

C.6 Operations or Group-Operations Users Report

```
92.086 16:21:16                                             RACF REPORT                                                 PAGE   1
                                                  USERS WITH OPERATIONS ATTRIBUTE
COMMAND GROUP ENTERED -
  RACFRW FORMAT TITLE('USERS WITH OPERATIONS ATTRIBUTE')
  SELECT PROCESS AUTHORITY(OPERATIONS)
  EV ALLSVC
  EV ALLCOMMAND
  LIST SORT(USER)
  END
92.086 16:21:16                              RACF REPORT - LISTING OF PROCESS RECORDS                                   PAGE   4
                                                  USERS WITH OPERATIONS ATTRIBUTE
                                                        E
                                                        V  Q
                                                        E  U
                     *JOB/USER  *STEP/    --TERMINAL--  N  A
 DATE    TIME   SYSID   NAME     GROUP       ID    LVL  T  L
92.083 22:06:34 XA01   STC       SYS1               0  7  0  JOBID=(DFHSM 92.083 06:57:47),USERDATA=(),OWNER=WAYNE
                           (NAME UNKNOWN)                    AUTH=(OPERATIONS),REASON=(CLASS)
                                                             SESSION=STARTED PROCEDURE,TOKEN STATUS=(
                                                             CREATED BY PRE 1.9 RACF CALL)
                                                             DATASET=HSM.BACK.T320622.WAYNE.ISPF.I2083,VOLUME=MIGRAT,LEVEL=00,
                                                             MODEL-ENTITY=WAYNE.ISPF.ISPPROF,MODEL-VOLUME=DSPAK1
92.083 22:06:55 XA01   STC       SYS1               0  7  0  JOBID=(DFHSM 92.083 06:57:47),USERDATA=(),OWNER=SYS1
                           (NAME UNKNOWN)                    AUTH=(OPERATIONS),REASON=(CLASS)
                                                             SESSION=STARTED PROCEDURE,TOKEN STATUS=(
                                                             CREATED BY PRE 1.9 RACF CALL)
                                                             DATASET=HSM.BACK.T500622.MXX.SYS1.I2083,VOLUME=MIGRAT,LEVEL=00,
                                                             MODEL-ENTITY=MXX.SYS1.CLIST,MODEL-VOLUME=MXACM2
92.083 22:10:23 XA01   STC       SYS1               0  7  0  JOBID=(DFHSM 92.083 06:57:47),USERDATA=(),OWNER=SYS1
                           (NAME UNKNOWN)                    AUTH=(OPERATIONS),REASON=(CLASS)
                                                             SESSION=STARTED PROCEDURE,TOKEN STATUS=(
                                                             CREATED BY PRE 1.9 RACF CALL)
                                                             DATASET=HSM.BACK.T151022.SYS1.CONFIG.I2083,VOLUME=MIGRAT,LEVEL=00,
                                                             MODEL-ENTITY=SYS1.CONFIG.IOCDS,MODEL-VOLUME=CCSGXX
92.083 22:10:31 XA01   STC       SYS1               0  7  0  JOBID=(DFHSM 92.083 06:57:47),USERDATA=(),OWNER=SYS1
                           (NAME UNKNOWN)                    AUTH=(OPERATIONS),REASON=(CLASS)
                                                             SESSION=STARTED PROCEDURE,TOKEN STATUS=(
                                                             CREATED BY PRE 1.9 RACF CALL)
                                                             DATASET=HSM.BACK.T241022.CCSG.VTAMLST.I2083,VOLUME=MIGRAT,LEVEL=00,
                                                             MODEL-ENTITY=CCSG.VTAMLST,MODEL-VOLUME=CCSGXX
92.083 22:11:32 XA01   STC       SYS1               0  7  0  JOBID=(DFHSM 92.083 06:57:47),USERDATA=(),OWNER=SYS1
                           (NAME UNKNOWN)                    AUTH=(OPERATIONS),REASON=(CLASS)
                                                             SESSION=STARTED PROCEDURE,TOKEN STATUS=(
                                                             CREATED BY PRE 1.9 RACF CALL)
                                                             DATASET=HSM.BACK.T291122.UCAT.INEWCAT.I2083,VOLUME=MIGRAT,LEVEL=00,
                                                             MODEL-ENTITY=UCAT.INEWCAT,MODEL-VOLUME=INFOPK
92.083 23:10:11 XA01   STC       SYS1               0  3  0  JOBID=(DFHSM 92.083 06:57:47),USERDATA=(),OWNER=STCSP4
                           (NAME UNKNOWN)                    AUTH=(OPERATIONS),REASON=(CLASS)
                                                             SESSION=STARTED PROCEDURE,TOKEN STATUS=(
                                                             CREATED BY PRE 1.9 RACF CALL)
                                                             DATASET=STCSP4.SPFTEMP1.CNTL,GENPROF=STCSP4.*,VOLUME=INFOPK,OLDVOL=
                                                             MXHSM3,LEVEL=00,ALLOWED=NONE
```

Appendix D. DSMON Reports Output Samples

This appendix contains examples of output reports using DSMON functions. All DSMON outputs have 133 columns. In this appendix the reports were truncated to best fit in the page.

D.1 Using Function SYSTEM

```
RACF DATA SECURITY MONITOR                                                         DATE: 03/26/92
                                   S Y S T E M     R E P O R T
-------------------------------------------------------------------------------------------------
CPU-ID                             310824
CPU MODEL                          9021
OPERATING SYSTEM/LEVEL             MVS/ 3.8   SP4.2.2 JBB4422
SYSTEM RESIDENCE VOLUME            MXARS1
SMF-ID                             XA01
RACF VERSION 1 RELEASE  9 IS ACTIVE
```

D.2 Using Function SYSPPT

```
RACF DATA SECURITY MONITOR                                                         DATE: 03/26/92
P R O G R A M    P R O P E R T I E S    T A B L E    R E P O R T
PROGRAM          BYPASS PASSWORD         SYSTEM
NAME             PROTECTION              KEY
-------------------------------------------------------------------------------------------------
AHLGTF                NO                 YES
HHLGTF                NO                 YES
IHLGTF                NO                 YES
PYGHJW                NO                 YES
ERBMIOPU              NO                 NO
ERBHYTGC              NO                 NO
IGGTGRF0              NO                 YES
IGDPPT01              YES                YES
THLMINIT              YES                YES
THLMISDO              NO                 YES
PLTSSDT               NO                 YES
RTWDNJST              NO                 YES
IOSPROUT              NO                 YES
ARFBHG01              NO                 YES
ARFBHG02              NO                 YES
ASBSCHWL              NO                 YES
ISSSPTRR              YES                YES
PPCTL                 NO                 NO
```

D.3 Using Function RACAUT

```
RACF DATA SECURITY MONITOR                                                         DATE: 03/26/92
R A C F    A U T H O R I Z E D    C A L L E R    T A B L E    R E P O R T
MODULE                RACINIT        RACLIST
NAME                  AUTHORIZED     AUTHORIZED
-------------------------------------------------------------------------------------------------
PGHHTL                NO             YES
```

D.4 Using Function RACEXT

```
RACF DATA SECURITY MONITOR                                                         DATE: 03/26/92
             R A C F    E X I T S    R E P O R T
EXIT MODULE          MODULE
NAME                 LENGTH
-------------------------------------------------------------------------------------------------
ICHPWX01             1,342
```

D.5 Using Function RACUSR

```
RACF DATA SECURITY MONITOR                                          DATE: 03/26/92
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 ----------------
               SPECIAL     OPERATIONS  AUDITOR      REVOKE
--------------------------------------------------------------------------------
ADRIANA                    SYSTEM
ANN            SYSTEM
AUDIT                                  SYSTEM
BILL           GROUP       GROUP
BMW            GROUP       GROUP       GROUP
DAVID                      SYSTEM      SYSTEM
DEMETRIO       GROUP       GROUP
DFHSM          SYSTEM      SYSTEM
DIANA          GROUP       GROUP       GROUP
DICK           GROUP       GROUP
EDWARD         SYSTEM
FDELROS        GROUP                   GROUP
GRAEFH                                              SYSTEM
HAIMO          SYSTEM      SYSTEM      GROUP
POWER          SYSTEM      SYSTEM      SYSTEM

RACF DATA SECURITY MONITOR                                          DATE: 03/26/92
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
--------------------------------------------------------------------------------
TOTAL DEFINED USERS:       267
TOTAL SELECTED ATTRIBUTE USERS:
ATTRIBUTE BASIS     SPECIAL        OPERATIONS       AUDITOR          REVOKE
---------------   --------------   --------------   --------------   --------------
SYSTEM                         5                5                3                1
GROUP                          6                5                4                0
```

D.6 Using Function RACSPT

```
RACF DATA SECURITY MONITOR                                          DATE: 03/26/92
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
PROCEDURE    ASSOCIATED   ASSOCIATED
NAME         USER         GROUP        PRIVILEGED  TRUSTED
--------------------------------------------------------------------------------
JES2         STC          SYS1           YES         NO
NET          STC          SYS1           YES         NO
NPM          STC          SYS1           YES         NO
TSO          STC          SYS1           YES         NO
DFHSM        STC          SYS1           YES         NO
RMF          STC          SYS1           YES         NO
LLA          STC          SYS1           YES         NO
SMF          STC          SYS1           YES         NO
LOGREC       STC          SYS1           YES         NO
RACFAUDT     AUDITOR      SYS1           YES         NO
ISPFLMF      ISPFLMF      TSO            NO          NO
SAMON        STC          SYS1           YES         NO
*            STC          TSO            NO          NO
```

D.7 Using Function RACCDT

```
RACF DATA SECURITY MONITOR                                          DATE: 03/26/92
R A C F    C L A S S    D E S C R I P T O R    T A B L E    R E P O R T
CLASS                                               DEFAULT     OPERATIONS
NAME        STATUS      AUDITING     STATISTICS     UACC        ALLOWED
--------------------------------------------------------------------------------
RVARSMBR    INACTIVE      NO            NO          NONE          NO
RACFVARS    INACTIVE      NO            NO          NONE          NO
SECLABEL    INACTIVE      NO            NO          NONE          NO
DASDVOL     ACTIVE        YES           YES         ACEE          YES
GDASDVOL    ACTIVE        YES           YES         ACEE          YES
TAPEVOL     ACTIVE        YES           YES         ACEE          YES
TERMINAL    ACTIVE        YES           YES         ACEE          NO
GTERMINL    ACTIVE        YES           YES         ACEE          NO
```

```
JESJOBS       ACTIVE        NO                NO           NONE              NO
APPL          ACTIVE        YES               YES          NONE              NO
TIMS          ACTIVE        YES               YES          NONE              NO
GIMS          ACTIVE        YES               YES          NONE              NO
AIMS          ACTIVE        YES               YES          NONE              NO
TCICSTRN      ACTIVE        YES               YES          NONE              NO
GCICSTRN      ACTIVE        YES               YES          NONE              NO
PCICSPSB      ACTIVE        YES               YES          NONE              NO
QCICSPSB      ACTIVE        YES               YES          NONE              NO
GLOBAL        ACTIVE        NO                NO           NONE              NO
GMBR          ACTIVE        NO                NO           NONE              NO
DSNR          ACTIVE        NO                NO           ACEE              NO
FACILITY      ACTIVE        NO                NO           NONE              NO
VMMDISK       INACTIVE      NO                NO           NONE              YES
VMRDR         INACTIVE      NO                NO           NONE              YES
VMCMD         INACTIVE      NO                NO           NONE              YES
VMNODE        INACTIVE      NO                NO           NONE              YES
FCICSFCT      ACTIVE        YES               YES          NONE              NO
HCICSFCT      ACTIVE        YES               YES          NONE              NO
JCICSJCT      ACTIVE        YES               YES          NONE              NO
KCICSJCT      ACTIVE        YES               YES          NONE              NO
```

D.8 Using Function RACGAC

```
RACF DATA SECURITY MONITOR                                                              DATE: 03/26/92
R A C F    G L O B A L    A C C E S S    T A B L E    R E P O R T
CLASS           ACCESS         ENTRY
NAME            LEVEL          NAME
-------------------------------------------------------------------------------------------------------
DATASET         READ           &RACGPID.*
                READ           &RACUID.*
                UPDATE         SYS1.BRODCAST
                READ           SYS1.HELP
RVARSMBR                       -- GLOBAL INACTIVE --
SECLABEL                       -- GLOBAL INACTIVE --
DASDVOL                        -- NO ENTRIES --
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 --
JESJOBS         ALTER          CANCEL.*.*.&RACUID%*
                READ           SUBMIT.*.RACUID%*.&RACUID
VMMDISK                        -- GLOBAL INACTIVE --
VMRDR                          -- GLOBAL INACTIVE --
VMCMD                          -- GLOBAL INACTIVE --
VMNODE                         -- GLOBAL INACTIVE --
VMBATCH                        -- GLOBAL INACTIVE --
SCDMBR                         -- GLOBAL INACTIVE --
FCICSFCT                       -- GLOBAL INACTIVE --
JCICSJCT                       -- GLOBAL INACTIVE --
DCICSDCT                       -- GLOBAL INACTIVE --
SCICSTST                       -- GLOBAL INACTIVE --
```

D.9 Using Function RACGRP

```
RACF DATA SECURITY MONITOR                                                            DATE: 03/26/92
R A C F      G R O U P     T R E E     R E P O R T
LEVEL   GROUP       (OWNER)
----------------------------------------------------------------------------------------------------

    1   SYS1        (DODELL  )
        |
    2   | @PL         (DODELL  )
        |
    2   | CCSG
        | |
    3   | | IHV         (VWHITE  )
        | |
    3   | | SLR         (DODELL  )
        | |
    3   | | SORT        (DODELL  )
        |
    2   | CIC321
        |
    3   | | DB2
        | | |
    4   | | | AUTH
        | | |
    4   | | | CSP
        | | |
    4   | | | DBD1
        | |
    3   | | DBD1
        |
    2   | CIC325
```

Index

Printed in U.S.A.

GG24-3120-01