Resource Access Control Facility

Security Administrator's Guide

Version 1 Release 10

SC28-1340-15

Resource Access Control Facility

Security Administrator's Guide

Version 1 Release 10

SC28-1340-15

Note

Before using this information and the product it supports, be sure to read the general information under “Notices” on page 335.

| **Sixteenth Edition (December 2005)**

This edition applies to Version 1 Release 10 of the Resource Access Control Facility (RACF), program number 5741-A05, and to all subsequent releases and modifications until otherwise indicated in new editions or technical newsletters.

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

IBM welcomes your comments. A form for readers' comments is provided at the back of this publication. If the form has been removed, address your comments to:

International Business Machines Corporation
Department 55JA, Mail Station P384
2455 South Road
Poughkeepsie, NY 12601-5400
United States of America

FAX (United States & Canada): 1+845+432-9405
FAX (Other Countries):
Your International Access Code +1+845+432-9405

IBMLink (United States customers only): IBMUSM10(MHVRCFS)
Internet e-mail: mhvrcfs@us.ibm.com
World Wide Web: http://www.ibm.com/servers/contact/

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

Contents

Figures

Tables

Preface

Purpose of This Document

This document contains information for Version 1 Release 10 of the Resource Access Control Facility (RACF), program number 5740-XXH.

RACF is a licensed program that provides security and repository services as follows:

- Identifying and verifying system users
- Authorizing access to resources (such as data) and functions (such as commands)
- Logging unauthorized attempts to enter the system
- Logging installation-selected attempts to gain access to protected resources

This document is intended for the *RACF security administrator* who is working on a VM operating system on which RACF is installed. This document helps the RACF security administrator do the following:

- Planning:
  - Decide how to use RACF to increase the security of the system.
  - Organize a security implementation team.
  - Plan whether to delegate administrative tasks (a decentralized approach) or to restrict administrative tasks to a few users (a centralized approach).
  - Plan which system resources to protect. For example:
    - System minidisks
    - SFS files and directories
    - Certain VM events
    - Terminals
  - Plan which user resources should be protected by security administrators and which should be protected by the users themselves. For example, RACF can protect the following user resources:
    - User minidisks
    - SFS files and directories
  - Plan which users and groups are to be known to RACF. For example, an installation can use RACF to require that all batch jobs be associated with a RACF-defined user ID.
- Daily administration:
  - Give users access to the system (assigning user IDs and passwords)
  - Give users access to system resources or functions
  - Assist users with access control problems (such as forgotten passwords or authorizations required to do their jobs).
- Coordinating with administrators of other products, such as the tape librarian.

Notes:

1. For information on using RACF to log activity on your system, including the setting of auditing controls and using the RACF report writer, see *RACF Auditor's Guide*.
2. For planning and migration information, see *RACF Migration and Planning* and the appropriate RACF program directory for your system.

Who Should Read This Document

This document is intended for security administrators, group administrators, and other administrators responsible for system data security and integrity on VM systems. In general, security administrators have the system-SPECIAL attribute, which allows them to issue any RACF command (except those that require the AUDITOR attribute). Group administrators have been granted specific authority (not usually granted to users) to perform security-related tasks related to a RACF group, or a RACF group's resources.

Note: This document should be read by RACF auditors, but is not the primary reference for them. RACF auditors should see *RACF Auditor's Guide*.

Readers must be familiar with the RACF concepts and terminology described in *RACF General Information*. The readers of this document should also be familiar with VM systems.

For additional information about developing a security plan, see the following documents:

- *VM/SP Introduction to Security*
- *RACF Auditor's Guide*.

How to Use This Document

Most of this document describes how to protect particular kinds of resources, such as minidisks, terminals, and SFS files and directories. In general, you will first need to define users to RACF and set some RACF options. Then, depending on your security plan, you will select which classes of resources to protect, and create resource profiles for them.

Where to Find More Information

The following sections describe where to find additional information about RACF VM.

RACF Version 1 Release 10 Publications

The publications in Table 1 on page xi contain detailed information about RACF Version 1 Release 10.

Table 1 (Page 1 of 2). The RACF 1.10 Library

Task	**Title**	**Order Number**	**Contents**
Evaluation Planning	*RACF General Information*	GC28-0722	Contains an overview of the product as a whole and highlights the new functions for the current release
Planning Installation Customization Diagnosis	*RACF System Programmer's Guide*	SC28-1343	Describes how to modify and maintain RACF
Installation Customization Diagnosis	*RACF Macros and Interfaces*	SC28-1345	Describes each product macro and its syntax and explains how to code the interfaces
Customization	*External Security Interface (RACROUTE) Macro Reference for MVS and VM*	GC28-1366	Describes the RACF system macros and explains how to code the interfaces
Planning Customization Administration	*RACF Security Administrator's Guide*	SC28-1340	Explains RACF concepts and describes how to plan for and implement RACF
Diagnosis	*RACF Diagnosis Guide*	GY28-1016	Explains how to diagnose problems in the RACF program product
Installation Customization Administration	*RACF Command Language Reference*	SC28-0733	Contains the functions and syntax of all RACF commands
Installation Customization Administration	*RACF Command Syntax Summary*	SX22-0014	Contains information extracted from *RACF Command Language Reference*
Administration Diagnosis	*RACF Messages and Codes*	SC38-1014	Contains the RACF messages, routing and descriptor codes, RACF manager return codes, and RACF-related system completion codes
Planning Customization Administration	*RACF Auditor's Guide*	SC28-1342	Describes auditing considerations as well as how to use the SMF data unload facility, the RACF report writer, and the data security monitor
End Use	*RACF General User's Guide*	SC28-1341	Explains how to perform common end-user tasks
Installation	*RACF Program Directory*	Shipped with the product	Describes how to install RACF

Table 1 (Page 2 of 2). The RACF 1.10 Library

Task	Title	Order Number	Contents
Planning Migration Installation Customization Administration Auditing Operation Application Development	*RACF Migration and Planning*	GC23-3054	Contains information to guide installations through the migration process from previous releases of RACF to RACF 1.10

Related Publications

The following publications contain additional information that may help you use RACF on your system. This document uses the following short titles to refer to these publications.

Table 2. Related Publications

Short Title	Full Title and Order Number	
Application Development Guide or *Application Migration Guide for CMS*	**z/VM (V3R1)**	*z/VM: CMS Application Development Guide*, SC24-5957
	z/VM (V4)	*z/VM: CMS Application Development Guide*, SC24-6002
Application Development Guide	**z/VM (V3R1)**	*z/VM: CMS Application Development Guide for Assembler*, SC24-5958
	z/VM (V4)	*z/VM: CMS Application Development Guide for Assembler*, SC24-6003
CMS Command Reference	**z/VM (V3R1)**	*z/VM: CMS Command Reference*, SC24-5969
	z/VM (V4)	*z/VM: CMS Command and Utility Reference*, SC24-6010
System Facilities for Programming	**z/VM (V3R1)**	*z/VM: CP Programming Services*, SC24-5956
	z/VM (V4)	*z/VM: CP Programming Services*, SC24-6001
System Codes or *System Messages*	**z/VM (V3)**	*VM/ESA: System Messages and Codes*, GC24-5974
	z/VM (V4)	*z/VM: System Messages and Codes - CMS*, GC24-6031
		z/VM: System Messages and Codes - CP, GC24-6030
		z/VM: System Messages and Codes - Other Components, GC24-6032
System Diagnosis Guide	**z/VM (V3)**	*VM/ESA: Diagnosis Guide*, GC24-5975
	z/VM (V4)	*VM/ESA: Diagnosis Guide*, GC24-6039
VM GCS Planning or *GCS Command and Macro Reference*	**z/VM (V3)**	*z/VM: Group Control System*, SC24-5951
	z/VM (V4)	*z/VM: Group Control System*, SC24-5998

Softcopy Publications

Information about RACF and your system is available on the following CD-ROMs. The CD-ROM online library collections include the IBM Library Reader, which is a program that enables you to view the softcopy documents.

SK2T-2067 *Online Library Omnibus Edition: VM Collection*

This collection contains the set of books for the z/VM and VM/ESA libraries; the files are available in BookManager and Portable Document Format (PDF) format.

SK2T-2180 *OS/390 Security Server RACF Information Package*

This softcopy collection kit contains the OS/390 Security Server (RACF) library. It also contains the RACF/MVS Version 2 product libraries, the RACF/VM 1.10 product library, product books from the OS/390 and VM collections, International Technical Support Organization (ITSO) books (Redbooks), and Washington System Center (WSC) books (orange books) that contain information related to RACF. The kit does not contain any licensed publications. By using this CD-ROM, you have access to RACF-related information for IBM products such as OS/390, VM/ESA, CICS, and NetView.

SK3T-4272 *z/OS Security Server RACF Collection*

This softcopy collection kit contains the RACF library for z/OS in both BookManager and Portable Document Format (PDF) files. You can view or print the PDF files with the Adobe Acrobat reader.

SK2T-2177 *IBM Redbooks S/390 Collection*

This softcopy collection contains a set of Redbooks pertaining to S/390 subject areas.

SK3T-7876 *IBM eServer zSeries Redbooks Collection*

This softcopy collection contains a set of Redbooks pertaining to zSeries subject areas.

Internet Sources

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

- Online library

  To view and print online versions of additional publications that may be helpful (for example, the latest editions of z/VM or z/OS publications), go to the following URL:

  `http://www.ibm.com/servers/eserver/zseries/zos/bkserv/`

- Redbooks

  The redbooks that are produced by the International Technical Support Organization (ITSO) are also available at the following URL:

  `http://www.ibm.com/redbooks/`

- RACF home page

  You can visit the RACF home page at the following URL:

  `http://www.ibm.com/servers/eserver/zseries/zos/racf/`

RACF Courses

IBM provides a variety of educational offerings for RACF. For more information about classroom courses and other offerings, refer to the following resources:

- See your IBM representative
- Read *Enterprise Systems Training Solutions,* GR28-5467
- Call 1-800-IBM-TEACH (1-800-426-8322)

To Request Copies of IBM Publications

Direct your request for copies of any IBM publication to your IBM representative or to the IBM branch office serving your locality.

There is also a toll-free customer support number (1-800-879-2755) available Monday through Friday from 8:30 a.m. through 6:00 p.m. Eastern Time. You can use this number to:

- Order or inquire about IBM publications
- Resolve any software manufacturing or delivery concerns
- Activate the program reorder form to provide faster and more convenient ordering of software updates

Summary of Changes

Summary of Changes for SC28-1340-15 for RACF Version 1 Release 10 for VM

This revision reflects the addition, deletion, or modification of information to support miscellaneous maintenance items, including:

- Trademarks
- Updated information regarding protecting Guest LANs in “Promiscuous Mode Authorization” on page 177.

Technical and editorial changes are indicated by a vertical line to the left of the change.

Summary of Changes for SC28-1340-14 for RACF Version 1 Release 10 for VM

New Information

- “Protecting Guest LANs and Virtual Switches” on page 176 describes the VMLAN class Guest LAN support for z/VM Version 5 Release 1.0 and higher.

Technical and editorial changes are indicated by a vertical line to the left of the change.

Summary of Changes for SC28-1340-13 for RACF Version 1 Release 10 for VM

This revision contains information in support of RACF Version 1 Release 10 for VM and miscellaneous maintenance updates.

Updated Information

- “Where to Find More Information” on page x contains updated publication information.

Technical and editorial changes are indicated by a vertical line to the left of the change.

Summary of Changes for SC28-1340-12 for RACF Version 1 Release 10 for VM

This revision reflects the addition, deletion, or modification of information to support miscellaneous maintenance items, including:

- Updates to the information in “Using the RPIDIRCT EXEC” on page 248
- A new section: “Recovery Process for ICHDIRMV” on page 269

Summary of Changes
for SC28-1340-11
RACF Version 1 Release 10 for VM

This revision contains changes to support RACF Version 1.10 for VM.

Information has been added that describes how to:

- Define shared user IDs
- Use the secured signon function
- Control security for OpenExtensions VM
- Control security for the VM shared file system (SFS), including how to:
  - Protect SFS files and directories
  - Control the use of SFS administrative and operator commands

Information about using RACF on MVS has been omitted from this edition.

Restructured database

RACF 1.10 operates only with the restructured RACF database. The nonrestructured database is no longer supported. Customers who have not restructured their databases should not use the information in the RACF 1.10 books. For information on restructuring the RACF database, see *RACF Migration and Planning* for RACF 1.9.0 or RACF 1.9.2.

Chapter 1. Introduction

Over the past few years it has become much easier to create and access computerized information. No longer is system access limited to a handful of highly skilled programmers; information can now be created and accessed by almost anyone who has taken just a little time to become familiar with the newer, easier-to-use, high-level inquiry languages. As a result of this improved ease of use, the number of people using computer systems has increased dramatically. More and more people are becoming evermore dependent on computer systems and the information they store in these systems.

As the general computer literacy and the number of people using computers has increased, the need for data security has taken on a new level of importance. No longer can the installation depend on keeping data secure simply because no one knows how to access the data. Further, making data secure does not mean just making confidential material inaccessible to those who should not see it; it means preventing the inadvertent destruction of files by people who may not even know that they are improperly manipulating data.

As the security administrator, it is your job to make sure your installation's data is properly protected. RACF can help you protect this data.

How RACF Meets Security Needs

The Resource Access Control Facility (RACF) licensed program can satisfy the preferences of the end user without compromising any of the concerns raised by security personnel. The RACF approach to data security is to provide an access control mechanism that:

- Offers effective user verification, resource authorization, and logging capabilities
- Supports the concept of user accountability
- Is flexible
- Has little noticeable effect on the majority of end users, and little or no impact on an installation's current operation
- Is easy to install and maintain

User Identification and Verification

RACF controls access to and protects resources on both multiple virtual storage (MVS) and virtual machine (VM) systems. For a software access control mechanism to work effectively, it must be able to first *identify* the person who is trying to gain access to the system, and then *verify* that the user is really that person.

RACF uses a *user ID* and a system-encrypted *password* to perform its user identification and verification. When you define a user to RACF, you assign a user ID and temporary password. The user ID identifies the person to the system as a RACF user. The password verifies the user's identity.

The temporary password permits initial entry to the system, at which time the person is required to choose a new password. Unless the user divulges it, no one else knows the user ID-password combination.

The secured signon function provides an alternative to the RACF password called a PassTicket, which allows workstations and client machines to communicate with a host without using a RACF password. Using this function can enhance security across a network. For more information, see Chapter 11, “Using the Secured Signon Function” on page 197.

Authorization Checking

Having identified a valid user, the software access control mechanism must next control interaction between the user and the system resources. It must authorize not only what resources that user may access, but also in what way the user may access them, such as for reading only, or for updating as well as reading. This controlled interaction, or authorization checking, is shown in Figure 1 on page 4. Before this activity can take place, however, someone with the proper authority at the installation must establish the constraints that govern those interactions.

With RACF, you are responsible for protecting the system resources, such as minidisks, terminals, and shared file system (SFS) files and directories, and for issuing the authorities by which those resources are made available to users. RACF records your assignments in *profiles* stored in the RACF database. RACF then refers to the information in the profiles to decide if a user should be permitted to access a system resource.

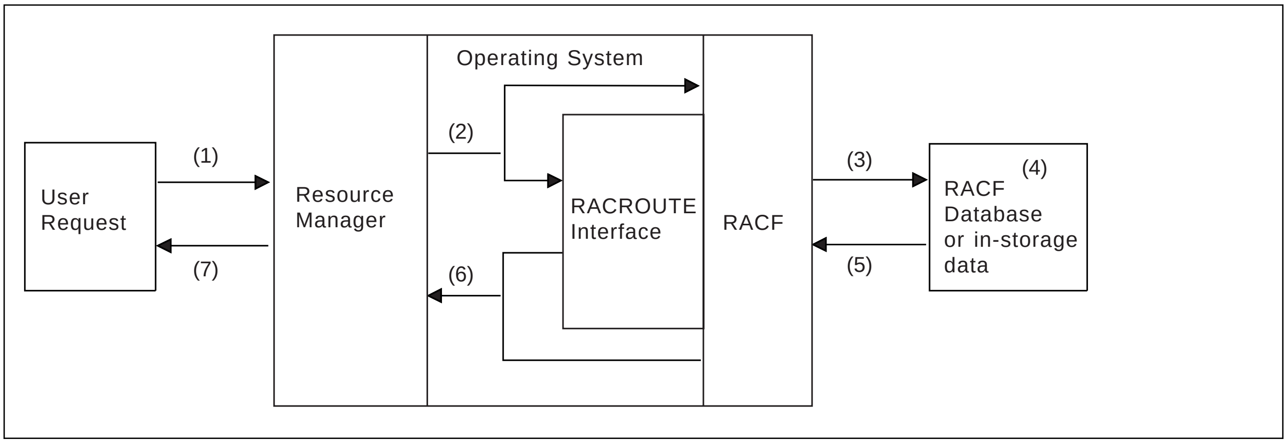


Figure 1. Authorization Checking

As shown in Figure 1, RACF performs authorization checking as follows:

1. A user requests access to a resource through a resource manager such as SFS or the z/VM control program (CP).
2. The resource manager issues a RACF request to find out if the user can have access to the resource. In most cases, this request is a RACROUTE macro; otherwise, it is an independent RACF macro.
3. RACF refers to the RACF database or to profiles copied into storage from the RACF database.
4. RACF checks the appropriate resource profile.
5. Based on the information in the profile...
6. ...RACF passes the status of the request—the user can or cannot have access—to the resource manager.
7. The resource manager grants or denies the user's request.

Logging and Reporting

The ability to log information, such as attempted accesses to a resource, and to generate reports containing that information can prove useful to a resource owner, and is very important to a smoothly functioning security system.

Because RACF can identify and verify a user's user ID and recognize which resources the user can access, RACF can record the events where user-resource interaction has been attempted. This function records actual access activities or variances from the expected use of the system.

RACF has a number of logging and reporting functions that allow a resource owner to identify users who attempt to access the resource. In addition, you and/or your auditor can use these functions to log all detected successful and unsuccessful attempts to access the RACF database and RACF-protected resources. Logging all access attempts allows you to detect possible security exposures or threats. The logging and reporting functions are:

- **Logging:** RACF writes records to SMF (system management facility) for detected, unauthorized attempts to enter the system. Optionally, RACF writes records to SMF for authorized attempts and/or detected, unauthorized attempts to:

  - Access RACF-protected resources
  - Issue RACF commands
  - Modify profiles on the RACF database.

  RACF writes SMF records to a CMS file. To list SMF records, you can use either the RACF report writer or the RACF SMF data unload utility (IRRADU00). With the report writer, you can select RACF SMF records to produce the reports. With the SMF data unload utility, you can translate the RACF SMF records into a format you can browse or upload to a database, query, or reporting package, such as SQL/DS.

  You should keep in mind that, for each logging activity that RACF performs, there is a corresponding increase in RACF and SMF processing. Also, the RACF report writer has been stabilized at the RACF 1.9.2 level and has not been enhanced for this release.

  For more information on logging and auditing, see *RACF Auditor's Guide*. For information on how to specify logging and auditing functions, see the *RACF Command Language Reference*.

- **Sending Messages:** RACF sends messages to the security console for detected, unauthorized attempts to enter the system and for detected, unauthorized attempts to access RACF-protected resources or modify profiles on the RACF database.

  As well as sending resource access violation messages only to the security console, RACF lets you send a message to a RACF-defined VM user. Each resource profile can contain the name of a user to be notified when RACF denies access to the resource. If the user is not logged on to the system at the time of the violation, the user receives a reader file that contains the notification information.

  If you are auditing access attempts, and if you have selected the RACF function that issues a warning message instead of failing an invalid access attempt (to allow for a more orderly migration to a RACF-protected system), RACF records each attempted access. For each access attempt that would have failed, RACF sends a warning message (ICH408I) to the accessor, but allows the access. If a “notify” user is specified in the resource profile, RACF also sends a message to that user. If you are deferring access authorization to VM through the use of the SYSSEC macro, and are auditing access attempts, RACF writes SMF records for access attempts that would have failed if you were not deferring.

- **Keeping Statistical Information:** Optionally, RACF can keep selected statistical information, such as the date, time, and number of times that a user enters the system and the number of times a single user accesses a specific resource. This information can help the installation analyze and control its computer operations more effectively. In addition, to allow the installation to track and maintain control over its users and resources, RACF provides commands that enable the installation to list the contents of the profiles in the RACF database.

User Accountability

Individual accountability should probably be one of your installation's prime security objectives. A user who can be held individually accountable for actions is less likely to make mistakes or take other actions that might disrupt or compromise operations at your installation.

When an individual VM user gains access to the system through a terminal, the concept of *individual user identity* is clear. With a group of production programs, however, it may be less clear just who the user is. (Is it the application owner, the job scheduling person, or the console operator?)

RACF offers you the ability to assign each user a unique identifier. (Of course, whether you establish this degree of accountability in all cases is an installation decision.)

In addition, RACF permits you to assign each user to one or more groups, which are simply collections of users having common access requirements.

RACF 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, on VM, a user ID can be associated with a disconnected service machine. In addition, in many systems today a “user” is equated with a function, rather than an individual. For example, a service bureau customer may comprise several people who submit work as a single user. Their jobs are simply charged to a single account number. From the security standpoint, as mentioned before, equating a user ID with anything other than an individual can be undesirable because individual accountability is lost. It is up to the installation, through you, to decide how much individual accountability is required.

RACF Groups

A RACF group is normally a collection of users with common access requirements. As such, it is an administrative convenience, because it can simplify the maintenance of access lists in resource profiles. By adding a user to a group, you can give that user access to all the resources that the group has access to. Likewise, by removing a user from a group, you can prevent the user from accessing those resources. You can also use groups as *holding* groups. For more information, see Chapter 4, “Defining Groups” on page 79.

The group concept is very flexible; a RACF group can be equated with almost any logical entity, such as a project, department, application, service bureau customer, operations group, or systems group. Further, individual users can be connected to any number of groups. Membership and authority in these groups can be used to control the scope of a user's activity.

What RACF Controls

On VM systems, you can use RACF to control access to:

- The system
- Terminals
- Minidisks
- OpenExtensions resources
- SFS files and directories
- Virtual unit record devices

- RSCS nodes
- A subset of CP commands, DIAGNOSE codes, and system functions
- Restricted segments
- Tape volumes (when a tape management system is installed)
- Printers
- Restricted segments
- Alternate user IDs (batch processing)
- Installation-defined resources

For more information, see “Protecting General Resources” on page 20.

How Users and Groups Are Authorized to Access Resources

Basically, a user's authority to access a resource while operating in a RACF-protected system at any time is determined by a combination of these factors:

- User's *identity*
- User's *attributes*
- User's *group authorities*
- *Security classification* of the user and the resource profile
- The *access authority* specified in the resource profile

Identity: When defining a user, the security administrator assigns a 1- to 8-character user ID. With this user ID, the user logs on to the system (or submits a batch job). When a user attempts to access RACF-protected resources, RACF uses the user ID to determine the user's access to those resources.

Attributes: The security administrator or a delegate can assign attributes to each RACF-defined user. The attributes determine various extraordinary privileges and restrictions a user has when using the system. Attributes are classified as either user-level attributes (or, simply, user attributes) or group-level attributes:

- User attributes: On VM systems, you can assign the SPECIAL, AUDITOR, OPERATIONS, CLAUTH, and REVOKE attributes at the system level. When you assign attributes at the system level, the privileges and restrictions apply across the entire system. See “Assigning Optional User Attributes” on page 16 and “User Attributes” on page 56 for detailed information about these attributes.

- Group-level attributes: When you assign an attribute at the group level, RACF limits the privileges or restrictions conveyed by the attribute to the group to which it applies (and to resources, users, and groups that fall within the scope of that group). See “Assigning Optional User Attributes” on page 16 and “User Attributes” on page 56 for more information about the group-SPECIAL, group-AUDITOR, and group-OPERATIONS attributes.

Group Authorities: Each user must be assigned (connected) to at least one group (called the user's default group). The security administrator or group administrator can assign a specific level of “group authority” to each user of a group. The group authorities are USE, CONNECT, and JOIN.

If a user has USE group authority within a group, the user can access resources to which the group is authorized.

CONNECT and JOIN also enable the user to access resources to which the group is authorized. However, these group authorities also give the user administrative responsibilities and privileges. The USE, CONNECT, and JOIN group authorities are described in detail in Chapter 4, “Defining Groups.”

Security Classification: Each user and each resource can have a security classification specified in its profile. The security classification can be a security level, one or more security categories, or both. A security *label* is an installation-defined name which refers to a combination of a security level and zero or more security categories. A security *level* is an installation-defined name that corresponds to a numerical security level (the higher the number, the higher the security level). A security *category* is an installation-defined name corresponding to a department or an area within an organization that has similar security requirements.

When a user requests access to a resource that has a security classification, RACF compares the security classification of the user with the security classification of the resource. For more information on security classifications, see Chapter 7, “Security Classification of Users and Data” on page 111.

Access Authority: The access authority determines to what extent the specified user or group can use the resource. The owner of a profile protecting a general resource (such as a tape volume or terminal) can grant or deny a user or group access to that resource by including the user ID or group ID in the resource profile's access list. Associated with each user ID or group ID is an access authority that determines whether the user or group can access the resource, and if they can access the resource, how they can use it.

The access authorities are NONE, EXECUTE, READ, UPDATE, CONTROL, and ALTER (see Table 46 on page 331).

For general resource profiles, an entry in the access list may also contain the name of a RACF-defined terminal. For more information, see “Conditional Access Lists for General Resource Profiles” on page 98.

Each resource also has a universal access authority (UACC) associated with it. The UACC can be NONE, EXECUTE, READ, UPDATE, CONTROL, or ALTER. The UACC is the access authority allowed to any user or group who is not represented in the access list. UACC applies to all users, whether they are RACF-defined or not.

RACF Profiles

As the security administrator or a delegate defines authorized users, groups, and protected resources, RACF builds *profiles*, which contain the information RACF uses to control access to the protected resources. Each profile is owned by a user or group. (By default, the owner of a profile is the user who creates it.)

You can work with the following types of profiles:

- User profiles
- Group profiles
- General resource profiles

User and group profiles contain descriptions of the authorized users of a RACF-protected system; general resource profiles contain descriptions of the resources and the levels of authority that are necessary to access these resources.

Flexibility

Because the security requirements at every data processing installation differ, RACF is designed to be flexible enough to assist each installation in meeting its own security objectives. There are a number of ways RACF accomplishes this:

- **Administrative Control:** RACF allows you a wide range of choices in controlling access to your installation's resources. RACF allows you to employ either centralized or decentralized administration techniques by permitting you to delegate authority, establish appropriate group ownership structures, and specify various group-related user attributes. In addition, RACF provides a wide range of processing options and installation exits.

  All RACF command functions, except those performed by the RVARY command, the RACFRW (report writer) command, and the BLKUPD command, have Interactive System Productivity Facility (ISPF) entry panels and associated help panels. These panels make it easy to enter command options on VM.

  You can define users and minidisks to both RACF and the VM directory using dual registration panels. For more information on dual registration panels, see “Using Dual Registration Panels” on page 296.

- **Generic Profiles:** RACF generic profiles allow you, your group administrators, and other users to define profiles that consolidate the security requirements of several similarly-named resources that have the same access requirements.

- **Protection of Installation-Defined Resources:** RACF allows you to protect your own installation-defined resource classes. To do this, add an entry in the class descriptor table (CDT) for the class of the resource, create profiles in the class, and, when a user requests access to a resource (or takes an action you wish to control), issue the RACROUTE REQUEST=AUTH macro from your application. You can control which users and/or groups can access each resource in the class by defining profiles in the class. The profiles can include access lists and other information such as auditing, security labels, and so forth, as with profiles in IBM-supplied classes. For more information on creating installation-defined resource classes, see *RACF System Programmer's Guide.*

- **Installation Exits:** RACF installation exits allow you to tailor RACF to specific needs of your installation. For more information, see “Using RACF Installation Exits to Customize RACF” on page 29.

Because of RACF's flexibility, you and your technical support personnel can tailor RACF to operate smoothly within the local operating environment.

Government Security Levels

The United States Department of Defense (DoD) has established security criteria for its computer systems and for those systems that perform government work under contract. Each system is evaluated and awarded a security rating, depending on the extent the system protects resources and its own processing. These ratings are, in ascending order of security, D, C1, C2, B1, B2, B3, and A1.

C2 Security Level

DoD describes a computing system that operates at the C2 security level as one that enforces discretionary access control by making users individually accountable for their actions through the use of logon procedures, auditing of security related events, and resource isolation. Such a system is called a **C2 trusted computing base**.

DoD has established the following criteria for operating a system at the C2 security level:

- **Discretionary Access Control (DAC):** Discretionary access control is a method of restricting access to resources (such as minidisks) based upon the identity of users or groups to which the users belong. DAC protects all system resources from unauthorized access down to a single user. A user who does not have permission to access a resource can only be granted this permission by the resource's owner.
- **Auditability of Security-Related Events:** Auditability of security-related events is the recording of facts that describe a security-related event in a computing system. These facts include the time and date of the event, the name of the event, the name of the system resources affected by the event, the name of the user who invoked the event, and so forth. For example, on VM systems, security-related events include certain CP commands, diagnose codes, system functions, and communication among virtual machines.
- **Resource Reuse:** Resource reuse is a practice that assures all system resources (such as tape data sets or minidisks) that are reused, reassigned, or reallocated are purged of all residual data, including encrypted data, belonging to the former owner.
- **Identification and Authentication:** Identification and authentication is a method of enforcing individual accountability by providing a way for each user to be uniquely identified. Users must then have their identity associated with any security-related, audited action they might take.

RACF 1.9.2 for VM with VM/ESA Version 1 Release 2.0 and a set of related products is designed to meet U. S. Department of Defense (DoD) C2 and B1 levels of security criteria as documented in *VM/ESA C2/B1 Trusted Facility Manual for VM/ESA with RACF*. New functions added to RACF since RACF 1.9.2 are also designed to meet C2 and B1 levels of security, with the following exceptions:

- When secured signon authentication is used from a remote node to authenticate to the VM host, the host system does not meet B1 or C2 criteria.
- RACF support of OpenExtensions for VM/ESA is designed to meet C2 security criteria.
- RACF support of the shared file system (SFS) is designed to meet C2 security criteria.

For more information about operating at a C2 level on VM, see *VM/ESA C2/B1 Trusted Facility Manual for VM/ESA with RACF*.

B1 Security Level

The requirements that the DoD sets forth concern two major areas: security policy and accountability. A security policy is a set of rules or practices that regulate how an organization handles its sensitive data. Accountability refers to the ability to establish a relationship between an action and the user responsible for the action.

Security Policy: The security policy that you implement in a B1 environment has as its major requirement a system of access controls that not only prevents individuals from accessing information at a classification that they are not authorized to, but also prevents individuals from declassifying information. The system must be able to protect resources of different levels of sensitivity.

The specific requirements are:

- **Mandatory Access Control (MAC):** Mandatory access control is a method of restricting access to resources based on the sensitivity of the information that the resource contains and the authorization of the user to access information with that level of sensitivity.

  You define the sensitivity of the resource by means of a label. This label indicates the level or classification of the information (for example, Restricted, Confidential, Internal). Within that level, you define which category the information belongs to (such as Project A, Project B). A user is able to access only that information in a resource to which the user's security label entitles it. If the user's security label does not have enough authority, then the user cannot access the information in the resource.

- **Discretionary Access Control (DAC):** Discretionary access control is the same for B1 as it is for C2.

- **Resource Reuse:** Resource reuse is the same for B1 as it is for C2.

- **Security Labels:** Security labels are associated with all users and resources in the system. The system uses these labels to determine if access to a resource is allowed under the mandatory access control (MAC) rules. Security labels, maintained in the RACF database, are usually defined by the security administrator and can be changed only by that person.

  When a resource is exported to a device attached to a system, the security label of the resource remains in effect. Whether the resource resides on a single-level device, such as a tape drive that does not process information at different levels of security concurrently, or a multilevel device, which is able to process data at different security levels concurrently, the system continues to associate the security label with the resource.

  The system provides security labels on each page of print output as a default. The system allows a user to request that no security labels be printed; however, the system is able to audit all such requests.

- **Identification and Authentication:** Identification and authentication is the same in B1 as it is for C2.

- **Auditing:** Auditing is the same for B1 as it is for C2, plus the following:

  - An audit record contains the audited resource's security label.
  - More selective options are available for audit reports.
  - The system can audit any override of labeling on printed output.

- **The Trusted Computing Base:** The “trusted computing base” is a “trusted computer system” that uses both hardware and software to ensure that these criteria are met. The security-relevant portion of the system, called the "trusted computing base," is made up of many separate hardware and software components. It is the combination of these components that enforce the security requirements of a system.

RACF 1.9.2 for VM with VM/ESA Version 1 Release 2.0 and a set of related products is designed to meet U. S. Department of Defense (DoD) C2 and B1 levels of security criteria as documented in *VM/ESA C2/B1 Trusted Facility Manual for VM/ESA with RACF*. New functions added to RACF since RACF 1.9.2 are also designed to meet C2 and B1 levels of security, with the following exceptions:

- When secured signon authentication is used from a remote node to authenticate to the VM host, the host system does not meet B1 or C2 criteria.
- RACF support of OpenExtensions for z/VM is designed to meet C2 security criteria.
- RACF support of the shared file system (SFS) is designed to meet C2 security criteria.

For more information about operating at a B1 level on VM, see *VM/ESA C2/B1 Trusted Facility Manual for VM/ESA with RACF*.

RACF Transparency

No users of a data processing system want their data destroyed or altered by other individuals (or by themselves) except when they specifically intend it. Unfortunately users of all types are often reluctant to take steps to protect what they have created. It is not uncommon to see live data used as test data, or to see data deliberately underclassified to avoid having to use the security procedures that the appropriate classification would demand. In many cases, people find it easier to ignore security procedures than to use them. Even conscientious users can forget to protect a critical piece of data. The solution to implementing effective security measures, then, is to provide a security system that is transparent (painless) to the user.

With RACF, end users need not be aware that their data is being protected for them. Security and group administrators can use generic profiles on both MVS and VM systems to make using RACF transparent to the majority of the installation's end users.

Administering Security

The security administrator's job can range from helping high-level management initially define corporate security policy to authorizing individual end users to access RACF-protected resources. As security administrator, you are responsible for implementing RACF at your installation. You have the authority to review and approve all implementation phases, select the resources to be protected, and plan the order in which protection will be implemented. You are the authority for all RACF implementation questions. You decide the degree to which decentralization of security controls takes place. You create profiles for the implementation team, select the team members, and direct their efforts.

Delegating Administration Tasks

While you have responsibility for overall security at your installation, you can decentralize much of the security operation by delegating various RACF security responsibilities to assistants. You can appoint:

- **Group Administrators:** Group administrators have many of the duties and responsibilities of a security administrator, but at a less inclusive level. Typically a group administrator will be responsible for defining the access requirements for the resources belonging to a single group. In some cases, the group administrator may delegate responsibilities in the same way as you delegated yours.
- **Technical Support:** The technical support person is typically a system programmer whose job is to install operating systems, apply fixes to problems in the operating systems, and write necessary programs to interface between operating system programs and application programs. The technical support person is responsible for providing you with technical assistance, installing and maintaining RACF, and for extending RACF to meet installation needs, as you direct. Technical support activities can include maintaining the RACF database.
- **Auditor:** The auditor supports the security implementation by ensuring that the levels of protection are adequate and that security exposures are reduced or eliminated. In addition, the auditor monitors operations to ensure that security procedures are being carried out properly.

In certain installations, it is possible that some of these functions might be combined. Further, the amount of delegation will vary from installation to installation. In some installations, there may be much delegation of authority, and there may be more than one technical support person or more than two levels of group administrators. Similarly, other roles may differ somewhat from the way they are described in this publication.

Using RACF Commands or Panels

After the planning for RACF implementation has taken place (see Chapter 2, “Organizing for RACF Implementation” for details), you can perform security and group administration tasks, largely, by using various RACF commands. For example, you can use the ADDGROUP command to define a new group as a subgroup of an existing group; you can use the ADDUSER command to define a new user and connect the user to the user's default group; you can use the ADDSD command to protect a DASD data set in an MVS environment, and so on. (Sample command sequences are given throughout this book for administrative tasks. See *RACF Command Language Reference* for the attributes and authorities you need to use RACF commands.) The RACF commands include operands with which you specify the various user attributes, group authorities, and access authorities. RACF places the information it receives from the commands into various profiles (data set profiles for MVS, and user, group, and general resource profiles for both VM and MVS), which it keeps in the RACF database and uses to control subsequent access to resources.

As an alternative to using RACF commands to perform administration tasks, you can use the ISPF panels (assuming that the ISPF product is installed at your location). If you use the panels, you need not memorize command or operand names; you need only fill in the appropriate information on the proper panels.

On VM systems, you can use dual registration to define users to both RACF and the VM directory at the same time. See "Using Dual Registration Panels" on page 296 for more information.

Choosing between Using RACF Commands and ISPF Panels

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

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

- Entering commands can be faster than displaying many panels in sequence.
- Using commands from book descriptions should be relatively straightforward. The examples in the books are generally command examples.
- Getting online HELP:
  - To see online help for the PERMIT command when you are using the RAC command, enter:

    ```
    RAC HELP PERMIT
    ```

  - In a RACF command session, enter:

    ```
    HELP PERMIT
    ```

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

    `HELP PERMIT SYNTAX` or `RAC HELP PERMIT SYNTAX`

- Getting message explanations (messages beginning with ICH or IRR):

  To see the explanation for message ICH06001I (a PERMIT message), enter:

  ```
  HELP PERMIT MSGID(ICH06001I)
  ```

  or:

  ```
  RAC HELP PERMIT MSGID(ICH6001I)
  ```

- If you use the RAC option, the RACF command output is displayed on the screen and also written to the RACF DATA file.

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

- ISPF creates a summary record in the ISPF log of the work you do. Unless you spool your console, the RACF commands do not create such a record. See *z/VM CMS User's Guide* for more information.
- From the panels, you can press the HELP key to display brief descriptions of the fields on the panels.
- The options chosen when installing the RACF panels will determine whether output (for example, profile listings, search results, RACF options, and VM event settings) is displayed in a scrollable form.

  If your installation uses XEDIT for display in ISPF, you can save the listings in a file. You can also save the output from a SEARCH in a REXX EXEC.
- The ISPF panels for working with VM events provide selection lists. Using the selection lists, you can avoid typing errors when specifying RACF event names.
- The ISPF panels for working with password rules allow you to enter all the password rules on one panel.

RACF Group and User Structure

Two of the fundamental elements of RACF are users and groups. Users, of course, are the many people who log on to a system, each with a unique user ID. In small installations, administration of a small number of users is not too difficult. However, when there are thousands of users, administration becomes a very large task. To make this task more manageable, the concept of groups was developed.

A group is a RACF entity with which any number of users are associated. Usually, the users in a group have some logical relationship to one another. The most frequent relationship is members of a department. Many installations pattern their group-user structure after their organization charts.

In the RACF group-user structure is an additional group called SYS1. When you install RACF, it will define this group for you. It is the highest group in the total RACF group-user structure. You can define your system administrator and system auditor as members of this group. The system administrator has the SPECIAL attribute, and the system auditor has the AUDITOR attribute. The significance of SPECIAL and group-SPECIAL and AUDITOR and group-AUDITOR, and the differences between them, are described in later sections.

Defining Users and Groups

You define users to RACF by issuing RACF commands that include various user attributes, as well as other control information RACF will use. The following are some of the commands you might use in your user-definition tasks. For a more complete description of the process of defining users, see Chapter 3, “Defining Users” on page 53. For complete descriptions of RACF commands, see *RACF Command Language Reference*.

Commands for User Administration

ADDUSER	Add a user profile to RACF.
ALTUSER	Change a user's RACF profile.
CONNECT	Connect a user to a group.
DELUSER	Delete a user profile from RACF and remove connection to all groups.
REMOVE	Remove a user from a group and assign a new owner for group data sets owned by the removed user.
LISTUSER	Display the contents of a user's profile.
PERMIT	Permit a user to access a resource (or deny access to a resource).
PASSWORD	Change a user's password.

In addition to defining individual users, you can define groups of users. Group members can share common access authorities to a protected resource.

One benefit of grouping users is that you can authorize the entire group, as a single unit, to access a protected resource. Another benefit is that attributes such as OPERATIONS can be assigned so that a given user has that attribute only when connected to a specific group, and the attribute is only effective for resources within the scope of that group.

The following are some of the commands you might use in your group-definition tasks.

Commands for Group Administration

ADDGROUP Define a new group (must be a subgroup of an existing group).

ALTGROUP Assign a subgroup to a new superior group.

DELGROUP Delete one or more groups.

LISTGRP Display the contents of a group profile.

CONNECT Connect a user to a group.

REMOVE Remove a user from a group and assign a new owner for group data sets owned by the removed user.

PERMIT Permit a group of users to access a resource (or deny them access to a resource).

Assigning Optional User Attributes

You can assign user attributes by specifying operands on RACF commands. User attributes describe various extraordinary privileges, restrictions, and processing environments that can be assigned to specified users in a RACF-protected system.

You can assign user attributes at either the system level or at the group level. When assigned at the system level, attributes are effective for the entire RACF-protected system. When assigned at the group level, their effect is limited to profiles that are within the *scope of the group*. The scope of control of a group-level attribute percolates down through a group-ownership structure from group to subgroup to subgroup, and so on. Percolation is halted (and therefore the scope of control of the group-level attribute) when a subgroup is owned by a user, rather than a superior group. Figure 2 on page 17 shows an example of the scope of control of an attribute assigned at the group level.

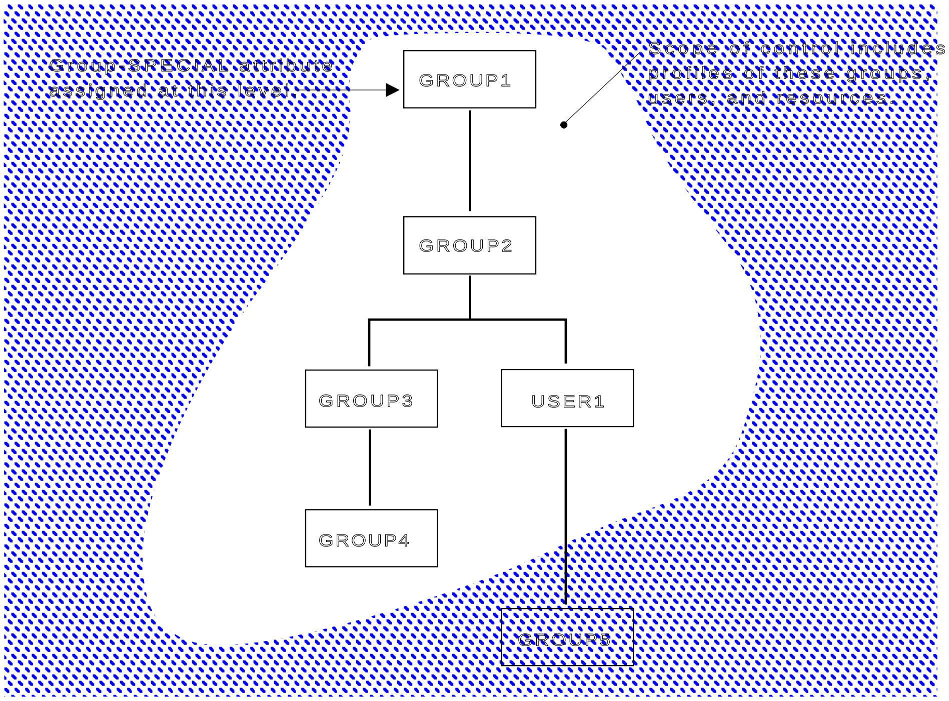


Figure 2. Scope of Control of an Attribute Assigned at the Group Level

Figure 2 shows a group ownership structure. In this figure, GROUP1 owns GROUP2, GROUP2 owns GROUP3 and USER1, and so on. A user who is connected to GROUP1 with the group-SPECIAL attribute has an explicit scope of control as shown in the figure. That is, the user cannot modify any profiles owned by GROUP5. Table 3 on page 18 lists and describes attributes that can be assigned at the user and group level. For a more complete description, see Chapter 4, “Defining Groups” on page 79.

Table 3. User Attributes

User Attribute	Description
SPECIAL	The SPECIAL attribute gives the user full control over all the RACF profiles in the RACF database when you assign it at the system level. At the system level, the SPECIAL attribute allows the user to issue all RACF commands. When you assign the SPECIAL attribute at the group level, the *group-SPECIAL* user has full control over all resources that are within the scope of the group, and cannot issue RACF commands that would have a global effect on RACF processing.
AUDITOR	When you assign the AUDITOR attribute at the system level, it gives the user full responsibility for auditing the security controls and the use of system resources across the entire system. With it, the user can specify logging options on the RACF commands, can list the auditing options of any profiles using the RACF commands, and can control additional logging to SMF for detecting changes and attempts to change the RACF database or for detecting accesses and attempted accesses of RACF-protected resources. When you assign the AUDITOR attribute at the group level (that is, when you assign the *group-AUDITOR* attribute), authority is restricted to resources that are within the scope of the group.
OPERATIONS	When you assign this attribute at the system level, it allows the user to perform any maintenance operations, such as copying, reorganizing, cataloging, and scratching, on RACF-protected resources. At the *group-OPERATIONS* level, authorization to perform these operations is restricted to the resources that are within the scope of the group.
CLAUTH	The CLAUTH (class authority) attribute allows the user to define profiles in a specific RACF class. A user can have class authority for the USER class and any of the classes defined in the class descriptor table. For example, IBM supplies the TERMINAL class (for terminals) and the TAPEVOL class (for tape volumes). For a list of valid class names, see “Protecting General Resources” on page 20. This authority is restricted if the SETROPTS GENERICOWNER option is in effect. See “Restricting the Creation of General Resource Profiles (GENERICOWNER Option)” on page 220.
GRPACC	When a user with the GRPACC attribute creates a DATASET profile for a group data set, RACF gives UPDATE access authority to other users in the group (if the user defining the profile is a member of that group.) A group data set is a data set whose high-level qualifier, or the qualifier derived from the RACF naming convention table, is a RACF-defined group name.
ADSP	The ADSP attribute establishes an environment in which all permanent DASD data sets created by this user are automatically defined to RACF and protected with a discrete profile. ADSP can be assigned at the group level, in which case it is effective only when the user is connected to that group.
REVOKE	This attribute excludes the RACF-defined user from entering the system. Revoke can be assigned at the group level, in which case the user cannot enter the system connected to that group.

Notes:

1. GRPACC and ADSP are MVS attributes. They have meaning in VM only if users with these attributes are using the VM system to maintain a RACF database that is shared between VM and MVS.
2. You and your delegates should assign the SPECIAL, AUDITOR, and OPERATIONS attributes to the minimum number of people necessary to administer security at the installation.

Assigning Group Authorities

Each user in a group may have different responsibilities for the group. These responsibilities may include creating resource profiles to be used by the group and adding new members to the group. You should assign a specific level of group authority to the user that is based on the user's responsibilities for administering and maintaining the group to which the user is connected. (You can do this with the ADDUSER, ALTUSER, or CONNECT command.)

The group authorities you can assign to a user are (in order of least to most authority): USE, CREATE, CONNECT, and JOIN. Each higher-level authority includes the lower levels of authority. Basically, the USE authority permits a user to access resources to which the group is authorized; the CREATE authority permits the user to create group data set profiles; the CONNECT authority enables the user to add previously RACF-defined users to the group; and the JOIN authority enables the user to define new users and new groups. See “Group Authorities” on page 84 for specific details.

Profiles Associated with Users and Groups

When you use the various RACF commands to define users and groups, the information RACF gathers from these commands is stored in profiles and placed in the RACF database. A general description of user and group profiles follows:

The User Profile: The user profile defines an individual user. Some of the things the user profile can contain are:

- Information about the user's identity, such as name and password (this can be masked, or encoded using RACF's implementation of the DES algorithm)
- System-wide user attributes
- The name of the user's default group
- Whether the user's security related activities should be logged
- How often the user's password is to be changed
- The user's security categories
- The user's security level
- The user's default security label

The Group Profile: The group profile defines a group. Some of the things the group profile can contain are:

- Information about the group, such as who owns it and what subgroups it has
- A list of connected users (members)
- The group authorities of each member

Protecting Resources

Enhancements to RACF, the VM operating systems and to applications have broadened the meaning of the term *resource* to include the following:

- Places in the system where data resides (such as minidisks on VM)
- Places in the system where data passes during data processing (such as terminals)
- The functions by which users work with data (such as commands)

Using RACF, you can protect resources so that only authorized users can access the resource in approved ways.

In general, you control access to a protected resource by creating a *discrete* or *generic* profile.

- *Discrete profiles* protect only one resource. The name of the profile identifies to RACF which resource is protected. On VM, a profile called SMITH.191 in class VMMDISK would protect SMITH's A-disk.
- *Generic profiles* protect one or more resources that have the same security requirements. In many cases, some of the characters in the resource names are the same. On VM, a profile called SMITH.* in class VMMDISK would protect all of SMITH's minidisks that did not have a more specific profile defined. SETROPTS GENLIST should be used when generic profiles are being used on VM. In most general resource classes, you can also provide a “top” generic profile that protects all resources not otherwise protected.

  Note: A top generic profile for a class should have a profile name of ** (rather than *) so that you can issue the RLIST command to display just the one profile.

Using generic profiles can greatly reduce the amount of RACF profile maintenance done by a RACF administrator.

Examples of discrete and generic profiles are shown throughout this book.

Protecting General Resources

To protect a general resource, create a general resource profile using the RDEFINE command. When you create a general resource profile, you must specify which general resource class the profile is in.

IBM-Supplied Resource Classes That Apply to VM Systems

DIRECTRY — Protection of shared file system (SFS) directories.

FACILITY — Miscellaneous uses. Profiles are defined in this class so resource managers (typically program products or components of MVS or VM) can check a user's access to the profiles when the users take some action. Examples are using combinations of options for tape mounts, and use of the RACROUTE interface.

RACF does not document all of the resources used in the FACILITY class by other products. For information on the FACILITY-class resources used by a specific product (other than RACF itself), see that product's documentation.

FIELD — Fields in RACF profiles (field-level access checking).

FILE — Protection of shared file system (SFS) files.

GLOBAL — Global access checking. [1]

GMBR — Member class for GLOBAL class (not for use on RACF commands).

GTERMINL — Terminals with IDs that do not fit into generic profile naming conventions. [1]

PSFMPL — When class is active, PSF/VM performs separator and data page labeling as well as auditing.

PTKTDATA — PassTicket Key Class.

PTKTVAL Used by NetView/Access Services Secured Single Signon to store information needed when generating a PassTicket.

RACFVARS RACF variables. In this class, profile names, which start with & (ampersand), act as RACF variables that can be specified in profile names in other RACF general resource classes.

RVARSMBR Member class for RACFVARS (not for use on RACF commands).

SCDMBR Member class for SECDATA class (not for use on RACF commands).

SECDATA Security classification of users and data (security levels and security categories).[1]

SECLABEL If security labels are used and, if so, their definitions.[2]

SFSCMD Controls the use of shared file system (SFS) administrator and operator commands.

TAPEVOL Tape volumes.

TERMINAL Terminals (TSO or VM). See also GTERMINL class.

VMBATCH Alternate user IDs.

VMBR Member class for VMEVENT class (not for use on RACF commands).

VMCMD Certain CP commands and other requests on VM.

VMEVENT Auditing and controlling security-related events (called VM events) on VM/SP systems.

VMLAN Use RACF to control Guest LANs

VMMAC Used in conjunction with the SECLABEL class to provide security label authorization for some VM events. Profiles are not allowed in this class.

VMMDISK VM minidisks.

VMNODE RSCS nodes.

VMRDR VM unit record devices (virtual reader, virtual printer, and virtual punch).

VMSEGMT Restricted segments, which can be named saved segments (NSS) and discontiguous saved segments (DCSS).

VXMBR Member class for VMXEVENT class (not for use on RACF commands).

VMXEVENT Auditing and controlling security-related events (called VM events) on z/VM systems.

VMPOSIX Contains profiles used by OpenExtensions VM.

WRITER VM print devices.

Notes:

1. You cannot specify this class name on the GENCMD, GENERIC, and GLOBAL/NOGLOBAL operands of the SETROPTS command.
2. You cannot specify this class name on the GLOBAL operand of the SETROPTS command or, if you do, the GLOBAL checking is not performed.

IBM-Supplied Resource Classes That Apply to OS/390 Systems

ALCSAUTH Supports the Airline Control System/MVS (ALCS/MVS) product

APPCLU Verifying the identity of partner logical units during VTAM session establishment.

APPCPORT Controlling which user IDs can access the system from a given LU (APPC port of entry). Also, conditional access to resources for users entering the system from a given LU.

APPCSERV Controlling whether a program being run by a user can act as a server for a specific APPC transaction program (TP).

APPCSI Controlling access to APPC side information files.

APPCTP Controlling the use of APPC transaction programs.

APPL Controlling access to applications.

CBIND Controlling the client's ability to bind to the server.

CONSOLE Controlling access to MCS consoles. Also, conditional access to other resources for commands originating from an MCS console.

CSFKEYS Controlling use of Integrated Cryptographic Service Facility/MVS (ICSF/MVS) cryptographic keys. See also the GCSFKEYS class.

CSFSERV Controlling use of Integrated Cryptographics Service Facility/MVS (ICSF/MVS) cryptographic services.

DASDVOL DASD volumes. See also the GDASDVOL class.

DBNFORM Reserved for future IBM use

DEVICES Used by MVS allocation to control who can allocate devices such as:

- Unit record devices (printers and punches) (allocated only by PSF, JES2, or JES3)
- Graphics devices (allocated only by VTAM)
- Teleprocessing (TP) or communications devices (allocated only by VTAM)

DIGTCERT Contains digital certificates and information related to them.

DIRAUTH Setting logging options for RACROUTE REQUEST=DIRAUTH requests. Also, if the DIRAUTH class is active, security label authorization checking is done when a user receives a message sent through the TPUT macro or the TSO SEND, or LISTBC commands. Profiles are not allowed in this class.

DLFCLASS The data lookaside facility.

DSNR Controlling access to DB2 subsystems.

FACILITY Miscellaneous uses. Profiles are defined in this class so that resource managers (typically program products or components of MVS or VM) can check a user's access to the profiles when the users take some action. Examples are catalog operations (DFP) and use of the vector facility (an MVS component).

RACF does not document all of the resources used in the FACILITY class by other products. For information on the FACILITY class resources used by a specific product (other than RACF itself), see the product's documentation.

FIELD Fields in RACF profiles (field-level access checking).

GCSFKEYS Resource group class for CSFKEYS class. [1]

GDASDVOL Resource group class for DASDVOL class. [1]

GLOBAL Global access checking table entry. [1]

GMBR Member class for GLOBAL class (not for use on RACF commands).

GSDSF Resource group class for SDSF class. [1]

GTERMINL Resource group class for TERMINAL class. [1]

IBMOPC Controlling access to OPC/ESA subsystems.

JESINPUT Conditional access support for commands or jobs entered into the system through a JES input device.

JESJOBS Controlling the submission and cancellation of jobs by job name.

JESSPOOL Controlling access to job data sets on the JES spool (that is, SYSIN and SYSOUT data sets).

LOGSTRM Reserved for MVS/ESA.

NODES Controlling the following on MVS systems:

- Whether jobs are allowed to enter the system from other nodes
- Whether jobs that enter the system from other nodes have to pass user identification and password verification checks

NODMBR Member class for NODES class (not for use on RACF commands).

OPERCMDS Controlling who can issue operator commands (for example, JES and MVS, and operator commands).[2]

PMBR Member class for PROGRAM class (not for use on RACF commands).

PROGRAM Controlled programs (load modules).[1]

PROPCNTL Controlling if user ID propagation can occur, and if so, for which user IDs (such as the CICS or IMS main task user ID), user ID propagation is *not* to occur.

PSFMPL Used by PSF to perform security functions for printing, such as separator page labeling, data page labeling, and enforcement of the user printable area.

PTKTDATA PassTicket Key Class enables the security administrator to associate a RACF secured signon secret key with a particular mainframe application that uses RACF for user authentication. Examples of such applications are IMS, CICS, TSO, VM, APPC, and MVS Batch.

RACGLIST Class of profiles that hold the results of RACROUTE REQUEST=LIST,GLOBAL=YES or a SETROPTS RACLIST operation.

RACFVARS RACF variables. In this class, profile names, which start with & (ampersand), act as RACF variables that can be specified in profile names in other RACF general resource classes.

RRSFDATA Used to control RACF remote sharing facility functions.

RVARSMBR Member class for RACFVARS (not for use on RACF commands).

SCDMBR Member class for SECDATA class (not for use on RACF commands).

SDSF Controls the use of authorized commands in the System Display and Search Facility (SDSF). See also GSDSF class.

SECDATA Security classification of users and data (security levels and security categories).[1]

SECLABEL If security labels are used, and, if so, their definitions.[2]

SERVER Controlling the server's ability to register with the daemon.

SMESSAGE Controlling to which users a user can send messages (TSO only).

SOMDOBJS Controlling the client's ability to invoke the method in the class.

STARTED Used in preference to the existing started procedures table to assign an identity during the processing of an MVS START command.

SURROGAT If surrogate submission is allowed, and if allowed, which user IDs can act as surrogates.

SYSMVIEW Controlling access by the SystemView for MVS Launch Window to SystemView for MVS applications.

TAPEVOL	Tape volumes.
TEMPDSN	Controlling who can access residual temporary data sets. You cannot create profiles in this resource class.
TERMINAL	Terminals (TSO or VM). See also GTERMINL class.
VTAMAPPL	Controlling who can open ACBs from non-APF authorized programs.
WRITER	Controlling the use of JES writers.
	CICS classes
ACICSPCT	CICS program control table. [2]
BCICSPCT	Resource group class for ACICSPCT class. [1]
CCICSCMD	Used by CICS/ESA 3.1, or later, to verify that a user is permitted to use CICS system programmer commands such as INQUIRE, SET, PERFORM, and COLLECT. [1]
CPSMOBJ	Used by CICSPlex System Manager, which provides a central point of control when running multiple CICS systems. Class CPSMOBJ will be used to determine operational controls within a CICSPlex.
CPSMXMP	Used by CICSPlex System Manager, which provides a central point of control when running multiple CICS systems. Class CPSMXMP will be used to identify exemptions from security controls within a CICSPlex.
DCICSDCT	CICS destination control table. [2]
ECICSDCT	Resource group class for DCICSDCT class. [1]
FCICSFCT	CICS file control table. [2]
GCICSTRN	Resource group class for TCICSTRN class. [2]
GCPSMOBJ	Resource grouping class for CPSMOBJ.
HCICSFCT	Resource group class for FCICSFCT class. [1]
JCICSJCT	CICS journal control table. [2]
KCICSJCT	Resource group class for JCICSJCT class. [1]
MCICSPPT	CICS processing program table. [2]
NCICSPPT	Resource group class for MCICSPPT class. [1]
PCICSPSB	CICS program specification blocks or PSBs
QCICSPSB	Resource group class for PCICSPSB class. [1]
SCICSTST	CICS temporary storage table. [2]
TCICSTRN	CICS transactions.
UCICSTST	Resource group class for SCICSTST class. [1]
VCICSCMD	Resource group class for the CCICSCMD class. [1]
	DB2 classes
DSNADM	DB2 administrative authority class
GDSNBP	Grouping class for DB2 buffer pool objects
GDSNCL	Grouping class for DB2 collection objects
GDSNDB	Grouping class for DB2 database objects
GDSNPK	Grouping class for DB2 package objects
GDSNPN	Grouping class for DB2 plan objects
GDSNSG	Grouping class for DB2 storage group objects
GDSNSM	Grouping class for DB2 system objects

GDSNTB	Grouping class for DB2 table, index, and view objects
GDSNTS	Grouping class for DB2 tablespace objects
MDSNBP	Member class for DB2 buffer pool objects
MDSNCL	Member class for DB2 collection objects
MDSNDB	Member class for DB2 database objects
MDSNPK	Member class for DB2 package objects
MDSNPN	Member class for DB2 plan objects
MDSNSG	Member class for DB2 storage group objects
MDSNSM	Member class for DB2 system objects
MDSNTB	Member class for DB2 table, index, and view objects
MDSNTS	Member class for DB2 tablespace objects

MVS/DFP and DFSMS/MVS classes

MGMTCLAS	SMS management classes.
STORCLAS	SMS storage classes.
SUBSYSNM	Authorizes a subsystem (such as a particular instance of CICS) to open a VSAM ACB and use VSAM Record Level Sharing (RLS) functions.

IMS classes

AIMS	Application group names (AGN).
CIMS	Command.
DIMS	Grouping class for Command.
FIMS	Field (in data segment).
GIMS	Grouping class for transaction.
HIMS	Grouping class for field.
OIMS	Other.
PIMS	Database.
QIMS	Grouping class for database.
SIMS	Segment (in database).
TIMS	Transaction (trancode).
UIMS	Grouping class for segment.
WIMS	Grouping class for other.

Information Management classes

GINFOMAN	Resource group class for Information Management Version 5.
INFOMAN	Member class for Information Management Version 5.

LFS/ESA classes

LFSCLASS	Controls access to file services provided by LFS/ESA.

MQM MVS/ESA classes

GMQADMIN	Grouping class for MQM administrative options. [1]
GMQCHAN	Reserved for MQM/ESA.
GMQNLIST	Grouping class for MQM namelists. [1]
GMQPROC	Grouping class for MQM processes. [1]
GMQQUEUE	Grouping class for MQM queues. [1]

MQADMIN	Protects MQM administrative options.
MQCHAN	Reserved for MQM/ESA.
MQCMDS	Protects MQM commands.
MQCONN	Protects MQM connections.
MQNLIST	Protects MQM namelists.
MQPROC	Protects MQM processes.
MQQUEUE	Protects MQM queues.
NetView classes	
NETCMDS	Controlling which NetView commands the NetView operator can issue.
NETSPAN	Controlling which NetView commands the NetView operator can issue against the resources in this span.
NVASAPDT	NetView/Access Services.
PTKTVAL	Used by NetView/Access Services Secured Single Signon to store information needed when generating a PassTicket.
RMTOPS	NetView Remote Operations.
RODMMGR	NetView Resource Object Data Manager (RODM).
z/OS UNIX System Services classes	
DIRACC	Controls auditing (via SETROPTS LOGOPTIONS) for access checks for read/write access to HFS directories. Profiles are not allowed in this class.
DIRSRCH	Controls auditing (via SETROPTS LOGOPTIONS) of HFS directory searches. Profiles are not allowed in this class.
FSOBJ	Controls auditing (via SETROPTS LOGOPTIONS) for all access checks for HFS objects except directory searches. Controls auditing (via SETROPTS AUDIT) of creation and deletion of HFS objects. Profiles are not allowed in this class.
FSSEC	Controls auditing (via SETROPTS LOGOPTIONS) for changes to the security data (FSP) for HFS objects. Profiles are not allowed in this class.
IPCOBJ	Controlling auditing and logging of IPC security checks.
PROCACT	Controls auditing (via SETROPTS LOGOPTIONS) of functions that look at data from, or affect the processing of, OpenExtensions VM processes. Profiles are not allowed in this class.
PROCESS	Controls auditing (via SETROPTS LOGOPTIONS) of changes to UIDs and GIDs of OpenExtensions VM processes. Controls auditing (via SETROPTS AUDIT) of dubbing and undubbing of OpenExtensions VM processes. Profiles are not allowed in this class.
z/OS DCE classes	
DCEUUIDS	Used to define the mapping between a user's RACF user ID and the corresponding DCE principal UUID
KEYSMSTR	Holds a key to encrypt the DCE password
TME 10 classes	
TMEADMIN	Maps the user IDs of TME administrators to RACF user IDs.
TSO classes	
ACCTNUM	TSO account numbers.
PERFGRP	TSO performance groups.

TSOAUTH	TSO user authorities such as OPER and MOUNT.
TSOPROC	TSO logon procedures.

Notes:

1. You cannot specify this class name on the GENCMD, GENERIC, and GLOBAL/NOGLOBAL operands of the SETROPTS command.
2. You cannot specify this class name on the GLOBAL operand of SETROPTS or, if you do, the GLOBAL checking is not performed.

Installation-Defined Classes

Your installation can add new class descriptors, or modify or delete existing class descriptors that you have added in the installation-defined part (module ICHRRCDE) of the RACF class descriptor table (CDT).

When you define a new resource class, you may optionally designate that class as either a resource *group* class or a resource *member* class. For a resource group class, each user or group of users permitted access to that resource group is permitted access to all members of the resource group. Note that for each resource group class you create, you must also create a second class representing the members of the group.

RACF refers to the class descriptor table whenever a class-related decision (such as, “What is the maximum length of profile names?”) must be made. With the services provided by the class descriptor table, and with appropriate use of RACF authorization checking services, you can extend RACF protection to any part of your system.

See the *RACF System Programmer's Guide* for more information on creating installation-defined classes.

Authority to Create Resource Profiles

Users can create FILE or DIRECTRY profiles if the second qualifier of the profile matches their user ID, or if they have the system-SPECIAL or group-SPECIAL attribute.

Users can create general resource profiles if they have the CLAUTH attribute for the class, or if they have the system-SPECIAL attribute.

Notes:

1. For complete descriptions of the required authorizations to any RACF command, see the description of the command in *RACF Command Language Reference*.
2. If the SETROPTS GENERICOWNER option is in effect, further restrictions apply. See “Restricting the Creation of General Resource Profiles (GENERICOWNER Option)” on page 220.

Authority to Modify or Delete Resource Profiles

To modify or delete a *generic* profile, you must meet at least one of the following criteria:

- Own the profile or, for FILE or DIRECTRY profiles, have a user ID that matches the second qualifier of the profile name.
- Have the SPECIAL attribute (or group-SPECIAL attribute, if applicable)

To modify a *discrete* profile, you must meet at least one of the following criteria:

- Own the profile or, for FILE or DIRECTRY profiles, have a user ID that matches the second qualifier of the profile name.
- Have the SPECIAL attribute (or group-SPECIAL attribute, if applicable)
- Have ALTER authority to the profile.

Note: For complete descriptions of the required authorizations to any RACF command, or if adding members, see the description of the command in *RACF Command Language Reference*.

Owners of Resource Profiles

In general, when you create a RACF profile, you are made the owner of the profile unless you specify otherwise. You can choose to specify either a RACF group or a RACF-defined user ID:

- If you make a *user* the owner of the RACF profile, the user can modify, list, and delete the profile, or name another user to become the owner.
- If you make a *group* the owner of a RACF profile, you extend the scope of the group (and, in some cases, the scope of its superior groups) to the RACF profile. If users have the group-SPECIAL, group-AUDITOR, or group-OPERATIONS attributes in these groups, their authority extends to the new profile. Further, if the profile is a group profile, the scope can extend to profiles owned by the group itself.

For a list of the RACF commands that owners of resource profiles can issue, see Table 47 on page 332.

Note: The concept of ownership of any kind of RACF profile (user, group, or resource) is different from other kinds of ownership:

- When a user attempts to access a protected resource, the user might be considered an “owner” of the resource, and be given the equivalent of ALTER access authority. This is true, for example, when a user reads an SFS file with a second qualifier that matches the user's user ID, or (unless the SECLABEL class is active) when a user attempts to LINK to his or her own minidisk (identified by the MDISK statement in the user's VM directory entry).

Setting up the Global Access Checking (GAC) Table

You can use global access checking to improve performance of RACF authorization checking for selected resources. For example, an entry in the global access checking table can allow all users on the system to have READ access to the MAINT 190 minidisk.

The global access checking table is maintained in storage and is checked early in the RACF authorization checking sequence. If an entry in the global access checking table allows the requested access to a resource, RACF performs no further authorization checking. This can avoid I/O to the RACF database to retrieve a resource profile, which can result in substantial performance improvements.

Note: If an entry in the global access checking table allows a requested access to a resource, no auditing is done for the request.

For information on planning and setting up the global access checking table, see Chapter 6, “Setting Up the Global Access Checking (GAC) Table” on page 103.

Security Classification of Users and Data

Security classification of users and data allows installations to impose additional access controls on sensitive resources. Each user and each resource can have a security classification in its profile. You can choose among the following:

- Security levels, security categories, or both.
- You can use security labels, which are a combination of security levels and security categories, and are easier to maintain.

For more information, see Chapter 7, “Security Classification of Users and Data” on page 111 and “B1 Security Level” on page 11.

Selecting RACF Options

RACF options provide flexibility in the creation and administration of your RACF security system. When implemented, RACF options can effectively enhance performance and recovery.

The SETROPTS command activates many of the RACF system-wide options. See Chapter 13, “Selecting RACF Options on VM (SETROPTS)” for a description of the SETROPTS options, as well as others.

Using RACF Installation Exits to Customize RACF

You can tailor RACF to bypass security checking or to perform additional security processing or checking by making use of various *installation exits*. (Installation exits are perhaps more in the realm of the technical support personnel, and as such are discussed in detail in *RACF System Programmer's Guide*. However, because you are responsible for overall security control at the installation, it is necessary for you to be aware of the use of installation exits.)

RACROUTE REQUEST=AUTH, DEFINE, VERIFY, and VERIFYX Exits

The RACROUTE REQUEST=VERIFY, REQUEST=AUTH, and REQUEST=DEFINE macros, respectively, perform user verification, access checking, and resource definition. Preprocessing installation exits are available to tailor the parameters specified by the RACROUTE REQUEST=VERIFY, REQUEST=AUTH, and REQUEST=DEFINE macro instructions or to perform any additional security checks. Postprocessing exits are available to override or modify results of RACF processing performed by these three macros. Because several of the RACF commands use RACROUTE REQUEST=AUTH and REQUEST=DEFINE when performing their functions, you can use the RACROUTE REQUEST=AUTH and REQUEST=DEFINE exits for some command tailoring.

RACROUTE REQUEST=LIST Exits

The RACROUTE REQUEST=LIST macro, used to build in-storage copies of general resource profiles, has two exits; the preprocessing/postprocessing exit and the selection exit. RACF calls the preprocessing/postprocessing exit before RACROUTE REQUEST=LIST processing to allow the installation to alter RACROUTE REQUEST=LIST processing options and after RACROUTE REQUEST=LIST processing to perform housekeeping. RACF calls the RACROUTE REQUEST=LIST selection exit to resolve conflicts between new and existing profile information.

RACROUTE REQUEST=FASTAUTH Exits

The RACROUTE REQUEST=FASTAUTH macro uses the resident profiles constructed by the REQUEST=FASTAUTH macro to perform authorization checking. The REQUEST=FASTAUTH macro has a preprocessing exit that allows you to make additional security checks or to instruct RACROUTE REQUEST=FASTAUTH to accept or fail the request to access a resource. The REQUEST=FASTAUTH macro also has a postprocessing exit that allows you to make additional security checks.

Command Exits

RACF provides exits that are called when the RACF commands ADDSD, ALTDSD, DELDSD, DELGROUP, DELUSER, LISTDSD, PERMIT, REMOVE, and SEARCH are issued. These exits permit the installation to perform additional authorization checking or to modify authorization checking when these commands are issued.

Password Processing Exit

The password processing exit supplements the processing RACF performs for new passwords and password change interval values. This exit gains control from the PASSWORD and ALTUSER commands and the RACINIT macro. The PASSWORD and ALTUSER commands and the RACINIT macro call this exit before actually changing the current password, the password change interval, or both.

RACF Encoding Exit

Using the ICHDEX01 exit, you can control how RACF either encodes (using a software implementation of the data encryption standard (DES) algorithm) or masks (using the RACF masking routine) the RACF password data that is stored in the RACF database. The ICHDEX01 exit also allows you to entirely replace the RACF DES encoding routine with whatever routine your installation chooses.

For information on coding RACF installation exits, see *RACF System Programmer's Guide*.

To see a report describing the RACF installation exits on your system, use the data security monitor (DSMON).

Tools for the Security Administrator

RACF provides a number of tools to help you (and the auditor) monitor and control RACF events.

Using the RACF Database Unload Utility

With a restructured database, you can unload the database to a sequential file by use of the database unload utility.

You can use the sequential file as input to a database management system, such as SQL/DS. You can then create a relational representation of the RACF database. Effective use of the relational database provides the security administrator with installation-tailored reports.

For information on running the utility, see “Using IRRDBU00 on VM (RACFDBU)” on page 135. For information on using the utility output, see “Using the Database Unload Utility Output” on page 141.

Using the RACF SMF Data Unload Utility

The RACF SMF data unload utility is the IBM-recommended utility for processing RACF audit records. With it, you can create a sequential file from the security-relevant audit data. The sequential file can be:

- Viewed directly
- Used as input for installation-written programs
- Manipulated with sort/merge utilities
- Uploaded to a data manager, such as SQL/DS, to process complex inquiries and create installation-tailored reports.

For information on how to use the SMF data unload utility, see *RACF Auditor's Guide.*

Using the RACF Report Writer

The RACF report writer lists information contained in RACF-generated SMF records. With the RACF report writer, you can:

- Collect data about successful accesses and warnings before building resource profile access lists.
- List the contents of RACF SMF records in a format that is easy to read.
- Obtain reports that describe attempts to access a particular RACF-protected resource. These reports contain the user ID, the number and type of successful accesses, the number and type of unauthorized access attempts, and the name of the user if available in the SMF record.
- Obtain reports that describe user and group activity.
- Obtain reports that summarize system and resource use.

The output from the RACF report writer includes a header page, which explains the meaning of the event and qualifier numbers that appear in SMF record listings and summary reports. The remainder of the report comes in various forms, according to your selection. You can request a general summary, SMF record listings, and summary reports.

You can find details on use of the report writer in *RACF Auditor's Guide*.

Using the Data Security Monitor

The data security monitor (ICHDSM00, usually called DSMON) is a batch program that allows authorized users to obtain a set of reports that provide information about the current status of your installation's data security environment. The reports that DSMON produces are:

- System report
- RACF exits report
- Selected user attribute report
- Selected user attribute summary report
- Class descriptor table report
- Global access table report
- Group tree report

These reports will help you to (1) check the initial steps you took to establish system security, and (2) make additional security checks periodically.

For more information on the DSMON reports, see “Using DSMON” on page 292 or *RACF Auditor's Guide*.

Recording Statistics in RACF Profiles

In addition to placing statistical information into the various profiles when you create them, you can cause RACF to dynamically record statistics (such as the number of user accesses to a protected resource) in discrete profiles. For general resource classes, you can optionally record statistics for the following:

- The number of times that a resource protected by a discrete profile was accessed under a specific RACF authority level (such as READ or UPDATE).

  Note: When a RACROUTE REQUEST=AUTH is accepted at a certain level of authority (such as UPDATE), this does not necessarily mean that data is actually updated.

- The number of times that a specific user or group accessed a resource protected by a discrete profile.
- The date when a resource profile was last updated.

These statistics enable you to monitor the current operation of your computing system for administrative and control purposes. You can list the statistics and other descriptive information recorded in RACF profiles with various RACF commands.

Note: Statistics are not recorded for the following:

- Generic profiles
- In-storage profiles (in classes for which the SETROPTS RACLIST command or the RACROUTE REQUEST=LIST macro have been issued)

Listing User IDs or Group Names Found in the RACF Database

You can list all occurrences of a user ID or group name in the RACF database, discover the relationships between various users and groups, and learn other important information about users, groups, and the resources they control by using the RACF database cross-reference utility (IRRUT100).

To invoke the RACF database cross-reference utility, you must be defined to RACF and must have the SPECIAL (or group-SPECIAL, as applicable) attribute. However, even if you do not have the SPECIAL or group-SPECIAL attribute, you can list occurrences of your own user ID.

This utility produces a cross-reference report that describes the occurrences of each user ID or group name you specify. Generic profile names are followed by the letter G in parentheses.

You can find complete information about the RACF cross-reference utility and other RACF utilities in *RACF System Programmer's Guide*.

As an alternative to IRRUT100, you can use the output from the database unload utility (IRRDBU00). Refer to “Using the Database Unload Utility Output” on page 141.

Listing Information from RACF Profiles

As shown in Table 4, You can use the LIST commands to list the contents of profiles.

To capture the output of RACF commands, do one of the following:

- In a RACF command session, spool your console.
- When using the RAC command processor, the output of RACF commands is placed in a file named RACF DATA on your disk or directory accessed as A, to which you have WRITE access. For information on how to change the defaults, see *RACF Command Language Reference*.

Table 4. Commands to List Profile Contents

Command	**Function**
LISTUSER	Lists the contents of user profiles. The listing shows the owner of the profile, the user name, the default group name, the groups that a user is connected to, group authorities, the date the password was last changed, the default security label, and other information. For a complete description of this listing, see *RACF General User's Guide*.
LISTGRP	Lists the contents of group profiles. The listing shows the owner of the group profile, the superior group name, the users connected to the group, the subgroup names, and other information.
LISTDSD	Lists the contents of DATASET profiles and allows you to determine which generic profile applies to a particular data set. The listing shows the owner of the profile, the UACC, the date the profile was created, the users and groups authorized to access the data set, your highest access authority to the data set, the security label (if there is one), and other information. **Note:** If you share a RACF database between VM and MVS, you can use this command while logged on to the VM system to list DATASET profiles in the database.
RLIST	Lists the contents of profiles for general resources such as tape volumes, VM minidisks, DASD volumes, and terminals. The listing shows the owner of the profile, the UACC, the date the profile was created, the users and groups authorized to access the resource, your highest access authority to the resource, the security label, and other information. For a description of this listing as it applies to minidisk profiles, see *RACF General User's Guide*.
LFILE	Lists the contents of an SFS file profile. For a description of this listing, see *RACF General User's Guide*.
LDIRECT	Lists the contents of an SFS directory profile. For a description of this listing, see *RACF General User's Guide*.

As an alternative to the LIST commands, you can obtain the contents of profiles from the output of the database unload utility (IRRDBU00). See “Using the Database Unload Utility Output” on page 141.

Searching for RACF Profile Names

You can list the names of profiles that meet certain search criteria by using the SEARCH command. This command is described in Table 5 on page 34.

Note: The output of this command is in line mode unless you use ISPF panels. On VM, you can use the RACF DATA file generated when you use the RAC command processor.

Table 5. Commands to Search for Profile Names

Command	Function
SEARCH	Searches the RACF database for the names of profiles (in a particular resource class) that match the criteria you specify. For example, you can search for all TERMINAL profiles that have a security level specified. You can save the list of profile names in a file. You can easily specify RACF commands (or other commands) to be saved with the profile names. On VM, you can use the ISPF panel dialog to to specify RACF commands (or other commands) to be saved with the profile names and to generate an EXEC.
SRFILE	Searches the RACF database for the names of profiles in the FILE class. This command is similar to the SEARCH command.
SRDIR	Searches the RACF database for the names of profiles in the DIRECTRY class. This command is similar to the SEARCH command.

When you use these commands, your search criteria can include one or more of the following:

- Profile names that contain a specific character string
- Profiles for resources that have not been referenced for more than a specific number of days
- Profiles that contain a level equal to the level you specify
- Profiles with the WARNING indicator
- Profiles that contain a specific security level
- Profiles that contain a specific security category
- Profiles that contain a specific security label
- Profiles to which another user has access.

Note: Unless you have the SPECIAL attribute, you must have at least READ access authority for each profile whose name is listed as the result of your request.

IRRDBU00 unloads your restructured database to a sequential file. The output from the database unload utility used with a relational database manager can provide you with the ability to implement additional search criteria.

Using RACF List and Search Commands Effectively

Attention

Using SEARCH can slow the system's performance. Therefore, SEARCH should be used with discretion (or not at all) during busy system times.

You may want to investigate using the database unload utility for some of your profile searches. The database unload utility need not slow the system's performance and, in some cases, may provide the same information as the SEARCH command.

Question: How can I tell if (or how) an SFS file or directory is protected?

Answer: Use the LFILE or LDIRECT command, omitting both the GENERIC and NOGENERIC operands:

LFILE *fn ft directory-name*

or

LDIRECT *directory-name*

This is documented in *RACF General User's Guide*.

Question: How can I tell if (or how) a resource is protected?

Answer: Use the RLIST command, omitting both the GENERIC and NOGENERIC operands:

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

Note: For resources that have grouping classes (such as terminals and DASD volumes), specify the related “member class” and the RESGROUP operand on the RLIST command:

```
RLIST member-class resource-name RESGROUP
```

For example, for terminal T1:

```
RLIST TERMINAL T1 RESGROUP
```

This lists the profiles in the GTERMINL class that protect terminal T1.

This example does not work for terminals protected by a generic member in the GTERMINL class.

Question: How can I find the SFS files and directories that a user can access?

Answer: Users can normally access all SFS files and directories in their own directory structure. In RACF this means that users have full authority to all profiles in the FILE and DIRECTRY classes that have a user ID qualifier that matches their own user ID. For example, user ID ANDREW has full authority to all FILE and DIRECTRY profiles which have ANDREW as the second qualifier.

Note: If SECLABEL protection is being used for the FILE and DIRECTRY classes, users may be restricted from accessing some of their own SFS files and directories, depending on the SECLABEL they are currently logged on with.

1. Find the names of the file and directory profiles the user has access to:

   ```
   SRFILE USER(userid)

   SRDIR USER(userid)
   ```

   The name of a discrete profile identifies which file or directory it protects.

   Note: If the SRFILE or SRDIR command shows a profile that contains a RACF variable (indicated by one or more & in the name) you must list the RACFVARS profile that defines the variable. For example if you see a directory profile named POOL1:ANDREW.&X.DATA, use the RLIST command to list the RACFVARS profile that defines the variable:

   ```
   RLIST RACFVARS &X
   ```

2. RACF provides no direct way to determine which SFS files or directories a particular generic profile protects, as in issuing the LISTDSD command with the DSNS operand for data sets. As described previously, you may use the LFILE or LDIRECT command to display the RACF profile which protects a particular SFS file or directory.

3. Find the entries in the global access checking table for the FILE and DIRECTRY classes:

   ```
   RLIST GLOBAL FILE

   RLIST GLOBAL DIRECTRY
   ```

   These entries allow all users access to files or directories that match.

Question: How can I find the general resources that a user can access?

Answer: This must be done one class at a time. For each class, take the following steps (similar to the steps for data sets):

1. Find the names of the profiles the user has access to:

   SEARCH CLASS(*class-name***) USER(***userid***)**

   The name of a discrete profile identifies which resource it protects.

   Notes:

   a. If the resource is in a class for which there can be resource group profiles (such as GTERMINL), issue the SEARCH command twice, once for the member class and once for the grouping class. For example, for terminals:

      SEARCH CLASS(TERMINAL) USER(*userid***)**

      SEARCH CLASS(GTERMINL) USER(*userid***)**

   b. If the SEARCH command shows a profile that contains a RACF variable (indicated by one or more & in the name) you must list the RACFVARS profile that defines the variable. For example if you see a profile named SAMPLE.&X.DATA, use the RLIST command to list the RACFVARS profile that defines the variable:

      ```
      RLIST RACFVARS &X
      ```

2. RACF does not provide a direct way to determine which resources a particular general resource profile protects. In general, there is no list stored on the system that shows the various existing resources RACF can check. There would have to be such a list for each general resource class—and there are well over 20 classes on VM alone, for such resources as terminals, VM minidisks, and SFS files. Thus, for any particular class, an auditor or administrator would have to consult with the profile owners (or system support) to determine exactly which resources a generic profile protects.

3. Find the entries in the global access checking table for the class:

   RLIST GLOBAL *class-name*

   These entries allow all users access to resources that match.

Question: How can I find the user or group profiles a user can list or alter?

Answer: Enter one of the following commands:

SEARCH CLASS(USER) USER(*userid***)**

SEARCH CLASS(GROUP) USER(*userid***)**

Question: How can I find out the members of a RACF group?

Answer: Enter the following command:

```
LISTGRP group-name
```

Question: How can I find out what groups a user belongs to?

Answer: Enter the following command:

```
LISTUSER userid
```

The output of these commands is described in more detail in the *RACF General User's Guide*.

EXECs Supplied for Use on RACF for VM

RACF for VM provides various EXECs on the product tape to facilitate installation, for use by the system administrators, auditors and general users, and for product maintenance.

To use an EXEC, type in the name of the EXEC and press the Enter key.

Auditing EXECs

EXEC **Description**

RACDSMON Executes the data security monitor (DSMON) program in the VM environment. See *RACF Auditor's Guide* for details of how to run this EXEC.

RACFADU Executes the RACF SMF data unload utility on VM. See *RACF Auditor's Guide* for more information.

RACRPORT Executes the RACF report writer. See *RACF Auditor's Guide* for more information.

SMFPROF Profile EXEC for the RACFSMF virtual machine which does the SMF switching and archiving tasks. See *RACF Auditor's Guide* for more information.

General Use EXECs

EXEC **Description**

ICHDIRMV Allows you to rename and relocate directories, subdirectories, and underlying files in the SFS environment. You can use this EXEC in place of the CMS RENAME DIRECTORY and CMS RELOCATE DIRECTORY commands when those commands are restricted. See “Using the ICHDIRMV EXEC” on page 267 for more information.

ICHSFSDF Works in conjunction with the RAC EXEC and is called when a RACF SFS command is issued. If the file pool ID or user ID has been omitted from an SFS format name, ICHSFSDF inserts the current file pool ID or the file space into the SFS format name.

You should not modify the ICHSFSDF EXEC.

ISPF Allows users to start an ISPF panel session, if ISPF is installed. The ISPF exec shipped is a sample (ISPFRACF EXECSAMP) that customers may tailor at installation time. See *RACF Program Directory* for more information.

RAC Allows users to enter RACF commands without entering a RACF command session. See *RACF System Programmer's Guide*.

RACFLIST Lists profiles for resources. See *RACF General User's Guide* for more information.

RACFPERM Grants or revokes authority to access resources. See *RACF General User's Guide* for more information.

RACGROUP Determines the ACIGROUP, if any, a user belongs to. See *RACF General User's Guide* for more information.

RACOUTP Displays the output of a RACF command that was issued with the RAC EXEC. It can also be used to tailor the output. The output is obtained from the RACF DATA file. See *RACF System Programmer's Guide* for more information.

RACSEC Displays the currently active security label for a user ID. For more information, see “Displaying the Current Security Label for a User ID” on page 121.

RCMDRFMT Tailors the syntax of RACF commands if they are entered with the RAC EXEC. For more information, see *RACF System Programmer's Guide*.

Installation EXECs

The following EXECs are used during installation of RACF.

EXEC **Description**

GENNUC Generates the modified CMS used on the RACF service machine.

RACALLOC Allocates space on an OS-formatted minidisk for use as a RACF database.

RACDSF OS-formats a minidisk for use as a RACF database.

RACINITD Initializes an OS-formatted minidisk for use as a RACF database.

RACONFIG Defines the configuration of the primary and backup RACF databases.

RACSETUP Contains FILEDEFs and ACCESS commands for all the RACF databases.

RPIBLDDS Builds or adds to a RACF database by executing the RACF statements generated by RPIDIRCT.

RPIDIRCT Scans the CP directory and produces RACF statements to build the RACF database.

See *RACF RACF Program Directory* for examples and more information.

For details on running RPIDIRCT to define OpenExtensions information, see “Using the RPIDIRCT EXEC” on page 248.

Security Administration EXECs

EXEC	Description
ICHSFS	Migrates SFS installations to RACF. See “Step 1: Migrate Existing SFS Authorities Into RACF Using ICHSFS” on page 271 for more information.
RACFDBU	Used to unload the RACF database to a sequential file. For a description of how to use RACFDBU, see “Using IRRDBU00 on VM (RACFDBU)” on page 135.
RACFDEL	Uses the output from IRRUT100 to generate commands designed to remove occurrences of a user ID from the RACF database. For a description of how to use RACFDEL, see “Running the RACFDEL and RPIDELU EXECs” on page 75.
RPIDELU	Used to execute the commands generated by RACFDEL. For a description of how to use RPIDELU, see “Running the RACFDEL and RPIDELU EXECs” on page 75.

System Programming EXECs

EXEC	Description
RACIPLXI	Automatically logs on the AUTOLOG2 virtual machine following RACF initialization. RACF only invokes this EXEC during initialization. It can be modified to issue messages appropriate to your installation.
RACFCONV	Updates the database templates. Executes the RACF utility IRRMIN00. See *RACF System Programmer's Guide* for more information.
RACFSVRS	Used to initialize multiple RACF service machines. See *RACF System Programmer's Guide* for more information.
RACSTART	Starts RACF. Issued from the RACF service machine.
RACSVRXI	Exercises control following an IUCV SEVER from CP on the CP to RACF path. The RACF service machine is the only machine that invokes this EXEC, but it can be modified to issue messages appropriate to your installation.
RACUT100	Lists all occurrences of a user ID or group name in a RACF database. Executes the RACF utility IRRUT100. See *RACF System Programmer's Guide* for more information.
RACUT200	Copies a RACF database and identifies inconsistencies in the database. Executes the RACF utility IRRUT200. See *RACF System Programmer's Guide*.
RACUT400	Splits, merges, and copies a RACF database. Executes the RACF utility IRRUT400. See *RACF System Programmer's Guide* for more information.

Chapter 2. Organizing for RACF Implementation

This publication describes the security administrator's tasks as they relate to RACF. A successful security program, however, goes well beyond the relationship of the security administrator to the software security program your company has chosen to protect its computerized data. This chapter discusses some of the early work you and other people must do before installing RACF.

Ensuring Management Commitment

Management's decision to install RACF will not, by itself, be enough to ensure adequate security at your location. Indeed, if management were to ignore security concerns after simply selecting *any* software protection package, the eventual result would most likely be failure of the security undertaking.

To be successful, a security implementation requires a management involved with questions of security policy, procedures, resources to be allocated to the security function, and accountability of users of the computer system. Without such management support, the security procedures will fall into disuse and will become more of an administrative chore than a viable protection scheme. (In fact, such a situation could breed a false sense of security that could lead to serious exposures.)

You should work with management to prepare a clear, inclusive statement of security policy. This statement should reflect:

- Corporate security policy
- Physical protection considerations
- Installation data processing security requirements
- User department security requirements
- Auditing requirements
- Statement of policy concerning outside users of the system
- Security attitudes expected from all users of the system

The resultant security policy will help to ensure that a security implementation team can prepare a RACF implementation plan that is both realistic and consistent with the installation's security policy.

Selecting the Security Implementation Team

To ensure a smooth implementation of RACF, careful planning is required, starting with your selection of an implementation team.

The implementation team should include the viewpoints of all of the user types (security and group administrators, auditor, technical support personnel, operations, and end user). In addition to knowing their own areas, the implementation team representatives should be familiar with, or have access to people who are familiar with, the following areas:

- RACF
- Privacy legislation
- Installation organization
- Installation standards
- Major application areas

As security administrator, you will lead the implementation team. For best results, you should keep the team as small as possible. You should ensure that the results of the team's work are reviewed and fully supported by management.

Responsibilities of the Implementation Team

Some of the responsibilities that might be assigned to the implementation team are:

- Defining RACF security objectives
- Deciding what to protect and how to report attempted violations
- Establishing resource ownership structures
- Developing the RACF implementation plan and installing RACF
- Educating all users of the RACF-protected system

A typical list of implementation team members and their responsibilities is shown in Table 6.

Table 6. Participants of the Implementation Team

User Type	**Responsibility**
Security Administrator	As security administrator, you have overall responsibility for RACF implementation. It is your job to ensure that the work of the implementation team is consistent with good security practice and in line with the security policy established earlier. (For example, on VM systems, you will need to have someone look at what security is currently defined in the VM directory and decide if that security is sufficient for your needs under RACF.) In addition, you or your delegate administrators should be responsible for educating the installation users about how RACF will be implemented. (That is, will there be a grace period before the new security procedures take effect? How will the implementation of RACF affect the day-to-day responsibilities of each user?)
Technical Support Person	The technical support person is normally a system programmer who installs RACF and maintains the RACF database. This person has overall responsibility for the programming aspects of system protection and provides technical input on the feasibility of implementing various aspects of the implementation plan. In addition, the technical support person writes, installs, and tests RACF exit routines, if they are required. If your installation has both MVS and VM and you are using RACF on both systems, you should ensure that technical personnel and other representatives of both systems are members of the implementation team. For more information, see *RACF System Programmer's Guide*.
Auditor	The auditor provides guidance on good auditing practice as it relates to data security and user access. This person implements the necessary RACF logging and reporting options to provide an effective audit of security measures. For more information on the auditor's duties, see *RACF Auditor's Guide*. This publication outlines the procedures that a system auditor should follow and describes the RACF report writer and the data security monitor.
User Representative	The user representative should be a prospective group administrator who represents a major application area—perhaps a user support services or liaison function.
Other Users	Other users might be considered as members of the implementation team if appropriate. For example: • On VM systems, the system administrator, an RSCS system programmer, or a PSF system programmer.

The rest of this chapter discusses some of the major responsibilities of the security implementation team.

Defining Security Objectives and Preparing the Implementation Plan

Working from the statement of security policy as a base, the implementation team prepares an *implementation plan*. This plan should answer the question “How do we get there from here?” Experience indicates that an evolutionary implementation of security, rather than a revolutionary one, is the most successful way to bring about adequate security measures in the quickest time possible.

The implementation team will need to set priorities about which data, applications, and users need to be secured. The implementation team should plan to phase in the security controls over a period of time to give the users a necessary period for adjustment.

The implementation plan should identify the major RACF events—when each must be completed, who will be responsible for each event, and interdependencies among events. In addition, the plan should take into account other significant activity planned during the same time period that could affect the implementation (new systems, hardware, applications, and so on). At an early stage it should also define a pilot group for whom protection of business data, jobs, and users, will be completed before undertaking protection of other business data. The pilot group provides a means of obtaining RACF experience before extending protection to the rest of the installation.

Deciding What to Protect

Every installation has varying amounts of confidential data and varying degrees of confidentiality. For example, a development laboratory might be primarily concerned with the confidentiality of new products, while a bank or an insurance agency would be concerned with the confidentiality of its customers' records. Generally speaking, though, all data falls into one of the following categories:

1. Very sensitive, confidential data, which requires protection from disclosure, modification, or destruction
2. Nonconfidential data, which is recoverable with little inconvenience if destroyed
3. The vast amount of data that falls between these two extremes, which should be protected from inadvertent or deliberate modification or destruction

The data in category 1 *must* be protected. What should also be considered is how to protect the data that *ought* to be protected in a simple yet effective manner—in a way that is transparent to the user of this data. The implementation team does a *risk evaluation* of the installation's data to determine which data needs what level of protection.

The task of protecting large quantities of data can take on significant proportions unless you can acquire this protection automatically. In the case of RACF, protecting data is quite simple and, once the controls are in place, practically free from administrative overhead.

Protecting Existing Data

To protect data that already exists on your system before RACF is installed, you will need to create RACF profiles. You can use either discrete or generic profiles. However, using generic profiles can reduce the administrative effort of this task, because one generic profile can protect many resources. For example:

- You can protect existing general resources (such as tape volumes or terminals) by using the RDEFINE command. If several resources in the same class have the same access requirements, you can use one profile to protect them. Not only does this save space, but it also saves administrative time.

  If the names of the resources contain some identical characters, you can usually create generic profiles with *, **, or % in the profile name to protect them.

  For certain classes, such as terminals, you can create resource grouping profiles to protect resources using names that do not lend themselves to the use of *, **, or %.

  For any general resource class, you can define a “RACF variable” that can be used in profile names in general resource classes. See “Choosing Among Generic, Resource Group, and RACFVARS Profiles” on page 93.

- On VM, you can protect existing minidisks by using the RDEFINE command to create profiles in the VMMDISK class. It is recommended that your installation protect existing minidisks when first installing RACF, by using the RPIDIRCT EXEC. The RPIDIRCT EXEC scans the CP directory and translates directory entries into RACF commands, including an RDEFINE command for each minidisk defined in the CP directory. This procedure is described in *RACF Program Directory.*

  You can use RPIDIRCT to scan the directory for OpenExtensions information. See “Using the RPIDIRCT EXEC” on page 248 for more information.

  You can also use ICHSFS to migrate existing SFS file and directory protection into RACF profiles. See “Step 1: Migrate Existing SFS Authorities Into RACF Using ICHSFS” on page 271 for more information.

Note: You will need to determine the appropriate UACC, access lists, and other information (such as security classification, if used) for each profile.

For resources that have unique security requirements, you will need to create discrete profiles.

Protecting New Data

To help you protect new data, RACF provides *generic profiles.* Use of generic profiles can decrease the amount of administrative effort because you can use a single generic profile to protect a large number of existing resources that have a similar naming structure. See “Protecting General Resources with Generic Profiles” on page 164 for more information.

Profile Modeling

Profile modeling enables RACF or an installation exit routine to copy information (such as the access list, owner, and logging options) from an existing profile when defining a new profile. (The copied profile is not necessarily identical to the model profile. See “Possible Changes to Copied Profiles When Modeling Occurs.”) This copying greatly reduces the effort needed to create new profiles. The following are some ways to use profile modeling.

- A user can use the FROM operand (and related operands) on the RDEFINE, ADDFILE, and ADDDIR commands to copy information from an existing profile into the new profile. RACF uses the specified profile as a model when creating the new profile. However, profile segment information is not copied to the new profile.
- A user can use the FROM operand (and related operands) on the PERMIT, PERMFILE, and PERMDIR commands to copy the access list from one existing profile into another existing profile.
- If the preceding methods are not sufficient, an MVS or VM installation can also use a RACROUTE REQUEST=DEFINE exit routine to supply either the name of a model profile or the profile itself.

Possible Changes to Copied Profiles When Modeling Occurs

When a profile is copied during profile modeling, the new profile could differ from the model in the following ways:

- RACF places the user creating the new profile on the access list with ALTER access authority or, if the user is already on the access list, changes the user's access authority to ALTER.
- If the model profile contains members (specified with the ADDMEM operand), the members are not copied into the new profile.
- If the SETROPTS MLS option is in effect, the security label, if specified in the model profile, is *not* copied. Instead, the user's current security label is used. For more information, see “Options Related to Security Labels” on page 233.

  Exception: When the SETROPTS MLS option is in effect, if the SETROPTS MLSTABLE option is also in effect and the user has the SPECIAL attribute, the security label specified in the model profile is copied to the new profile. For more information, see “Options Related to Security Labels” on page 233.
- For TAPEVOL profiles, TVTOC information is not copied to the new profile.

Allowing a Warning Period

In addition to deciding what to protect, the implementation team will need to consider how to phase in the new security controls with minimum disruption of current work patterns. You should consider auditing all accesses allowed by a resource profile (specifying GLOBALAUDIT(ALL) for the resource profile) or auditing all protected resources in a class (entering the SETROPTS LOGOPTIONS command). These commands will cause SMF logging to occur for all accesses. If the profiles allow all access, the SMF records will indicate what users (or jobs) need access to the protected resources.

RACF also provides the option of issuing a warning message to users instead of failing a request to access a resource. You can control which resources are protected in this manner by indicating on the ADDSD, RDEFINE, ADDFILE, ADDDIR, ALTDSD, RALTER, ALTFILE, or ALTDIR command that a WARNING is desired. When a resource check is performed, if the check fails and WARNING has been specified, RACF issues a warning message to the user, logs the access, and allows the user access to the resource.

Note: The warning message facility applies to in-storage profiles created by the SETROPTS RACLIST command. It may or may not apply to in-storage profiles created by the RACROUTE REQUEST=LIST command, depending on the options chosen by the resource manager issuing the macro.

On VM, in addition to using the warning message facility, you can use the SYSSEC macro to defer access authorization decisions to VM for some frequently used resource classes until you have created profiles for those resources. The SYSSEC macro is coded in RACF module HCPRWA. For more information on operating your system in this way, see the description of the SYSSEC macro in *RACF Macros and Interfaces*.

Establishing Ownership Structures

RACF provides enough flexibility so that in most cases your RACF ownership structures can correspond to your existing installation management and organizational structures; however, this flexibility does not mean that some realignment of the organizational structures might not be advantageous from the security standpoint.

In any event, you should subdivide the ownership structures to minimize both occasions when data needs to be passed between groups, and occasions when exceptional access controls are required. If you define groups so that all users in a group share common access requirements, your administrative task of authorizing users is greatly simplified.

Selecting User IDs and Group Names

In your installation it might be enough for you to simply isolate development work from production. On the other hand, it might be more practical for you to define many individual users and groups. In either case, you should take a look at what already exists and modify RACF to adapt to the current environment. For example, do any or all of the system users already have user IDs? If so, perhaps you can make use of them.

Whatever you choose, consider carefully the longer term security objectives: Adding new groups and users to an existing structure presents few administrative problems; even deleting users and groups can be done without much difficulty. However, a major reassignment of user IDs and group names, while possible, is best avoided by careful initial selection.

Establishing Your RACF Group Structure

You should map your groups to your organization's structure and arrange them hierarchically (with the IBM-supplied SYS1 group as the highest group), so that each group is a subgroup of some other group. You should document the resulting group structure as part of the implementation plan. Perhaps you might want to develop a set of guidelines for your delegated security and group administrators to identify the general categories of resources, users, and the relationships between them.

Figure 3 shows relationships that can exist between users and groups.

Note: Although this figure uses MVS terminology to illustrate relationships between users and groups and their resources, the same concepts apply to relationships between users and groups and their resources on VM systems.

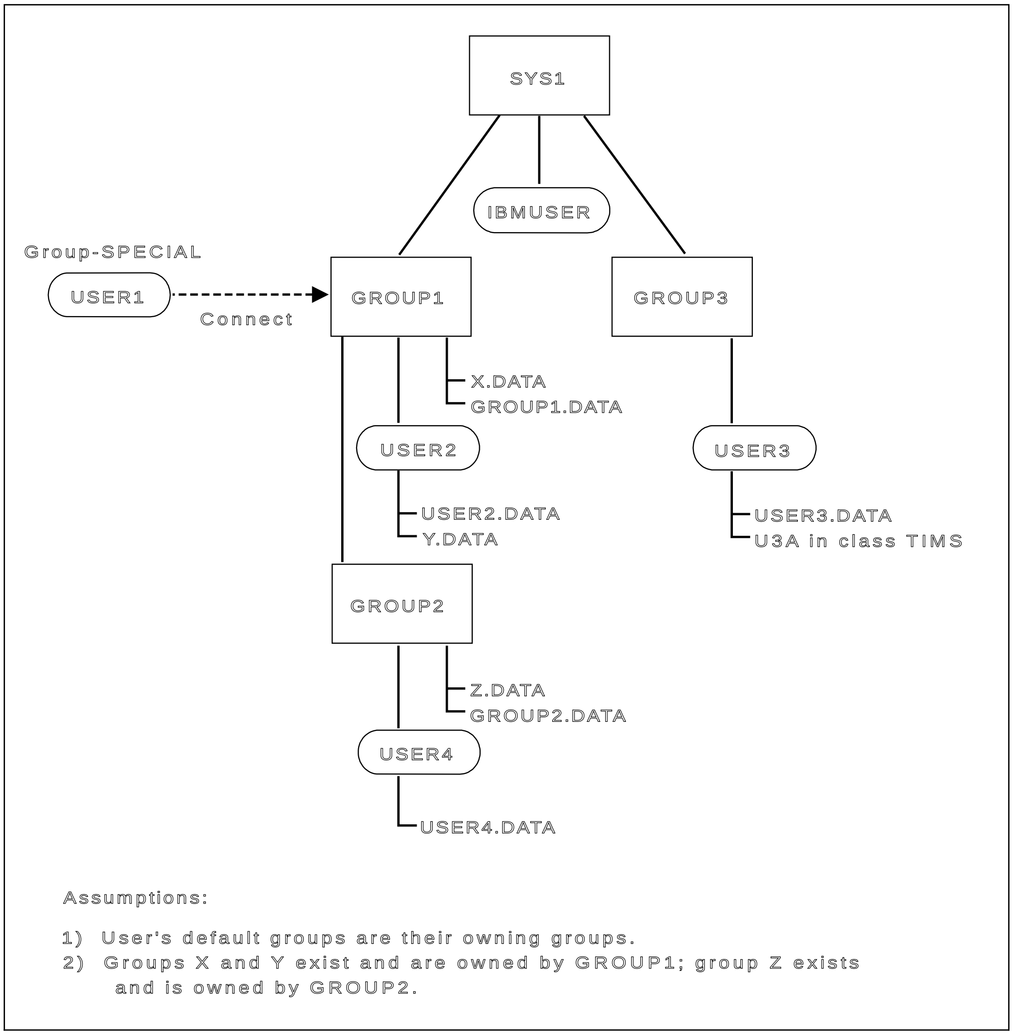


Figure 3. User and Group Relationships

In Figure 3, the highest level group, SYS1, owns subgroups GROUP1 and GROUP3 and the user, IBMUSER. GROUP1, in turn, owns subgroup GROUP2 and the users USER1 and USER2. Note that USER1 is connected to GROUP1 with group-SPECIAL authority. This gives USER1 (who is a RACF administrator) control over GROUP1's resources and resources in GROUP1's scope, but not GROUP3's resources.

Note: If you run with list-of-groups checking inactive (such as, SETROPTS NOGRPLIST), then the scope of USER1's group-SPECIAL attribute is limited to his default group or the group he specified when logging on, and the groups below that group in the hierarchy. For more information on list-of-groups checking, see “List-of-Groups Authority Checking” on page 219.

Educating the System Users

Part of your job is to tell the system users what they need to know to work without disruption when RACF is installed.

The amount of detailed information each user needs to know about RACF depends on the RACF functions you authorize the person to use. Some examples of information required by various types of system users are:

All System Users: All users defined to RACF must know the following:

- How to identify themselves to the system with their user ID and password.
- How to change their password.

  Users might need to be familiar with the RACF PASSWORD command if you want them to be able to change their password intervals.
- The significance of their password to system security.

  VM users and administrators should be aware that, when logging on with RACF, a user can change a current password or establish a new password when the old one has expired. Prior to installing RACF, these options were not available to the user.

Make sure to inform all users that they can use either the RAC command processor or the RACF command session. In addition, users should be able to:

- Use the RACF LISTUSER command to list their own profile information
- Protect their own VM minidisks with RACF profiles
- Protect their own SFS files and directories with RACF profiles
- Protect their own OpenExtensions resources as documented in *OpenExtensions for z/VM: User's Guide.*

If security labels are used on your system, users should know how to log on with a security label other than their default security label. For more information, see “How Users Specify Current Security Labels” on page 121.

Note: Users can enter the following command to find out what security labels they can use:

```
SEARCH CLASS(SECLABEL)
```

For complete information on tasks general users might perform using RACF, such as permitting others to use their minidisks, see *RACF General User's Guide*.

Users Who RACF-Protect General Resources: Depending on your security plan, users might work with profiles in the TAPEVOL or other general resource classes. These users must know:

- How to define and modify profiles in the general resource class, including whether generic profiles are allowed in the class
- What user IDs and group IDs they can use when giving access to the profiles

- The meaning of the access authorities (NONE, READ, and so forth) in the general resource class
- What your installation's security policy is towards specific security enhancements like security levels, categories, and security labels.

For more information, see Chapter 5, “Defining Profiles for General Resources” on page 89 and the sections of this book that describe how to use the class.

Technical Support Personnel: Users who install RACF need to be familiar with migration considerations and the steps required to install or re-install RACF. See *RACF Program Directory*.

Users who maintain the RACF database (for example, technical support personnel) must be familiar with the RACF utilities. *RACF System Programmer's Guide* describes these utilities.

Group Administrators: Group administrators have one of the group authorities, a group attribute (such as group-SPECIAL), or own group resources. These users will need to have information described in this book and in *RACF Command Language Reference*.

RACF Auditors: Users with the AUDITOR attribute should refer to *RACF Auditor's Guide* for information on using RACF for auditing.

Note that if ISPF is installed, the user can use the RACF ISPF panels to perform the same functions as the RACF commands. Using the RACF ISPF panels frees users from the need to know the details of command syntax.

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

RACF Command Language Reference contains detailed information on the RACF commands.

Programmers Writing Unauthorized Applications: Programmers writing unauthorized applications can use the RACROUTE macro in their applications to request many security-related services, including controlling access to protected resources (RACROUTE REQUEST=AUTH).

Note: Your installation can create installation-defined resource classes. If your installation creates profiles in those classes, an application can issue a RACROUTE REQUEST=AUTH to check if a user has sufficient authority to complete a user action. How much authority is needed for any particular user action is defined by the way the application invokes the RACROUTE REQUEST=AUTH macro. For more information on creating installation-defined classes, see *RACF System Programmer's Guide*.

Programmers Writing Authorized Applications: Programmers writing authorized applications can use the RACROUTE macro in APF-authorized programs to request security-related services, including:

- User identification and verification (RACROUTE REQUEST=VERIFY)
- Replacing or retrieving fields in RACF profiles (RACROUTE REQUEST=EXTRACT)

For more information on using the RACROUTE macro, see *External Security Interface (RACROUTE) Macro Reference for MVS and VM*.

Checklist for Implementation Team Activities

As an overall strategy in organizing for RACF implementation, the implementation team should strive for a policy of security by evolution, rather than revolution. Wherever transparency can be used, it should. In some cases, you will have to actively solicit management support.

You should examine organizational structures to establish the most efficient profile ownership structures, educate users with the level of material they need to perform their assigned functions, and prepare guidelines for the various administrators.

Finally, you and the implementation team should prepare an implementation plan to reflect the work of the team. Table 7 on page 52 provides a checklist for the implementation team to use while preparing the implementation plan. Note that this checklist represents only a starting point; it is not meant to be exhaustive.

Table 7. Checklist for Implementation Team Activities

Item	Comments
Objectives	What are the installation's security objectives? Over what time frame are they to be achieved? Is management's position clear on all objectives? Is the statement of security policy clear and complete for all objectives?
Protection	What resource classes are to be protected? Which resources within these classes are to be protected? Can protection be phased in? Which resources need to be protected when?
Naming conventions	What installation data set or general resource naming conventions exist? Are changes necessary? Will implementing RACF provide an opportunity to enforce naming conventions? And if so, installation-wide, or just a subset? Immediately or eventually?
Organization	Can the definition of RACF groups (and their associated users) be molded to map into the existing organizational structure? What changes to the organizational structure, if any, are necessary? How is RACF to be controlled and administered? Which functions are to be retained centrally? Which are to be delegated, wholly or in part? Which users should have what RACF attributes?
User and group names	Establish names for groups and user IDs. Determine which are to be defined to RACF. Select the user verification technique.
Transparency	Try to make RACF transparent to your users wherever possible. Consider which resources can be protected by generic profiles and which resources will require discrete profiles. Determine which users and groups should be placed in the access lists, and with what access authorities. Determine what deviations from strict user accountability are to be allowed, and for how long?
RACF tailoring	Determine which RACF exits are to be used, if any, and under what conditions.
VM authorizations	Review VM authorizations, such as: • Minidisk links • Access to RSCS network nodes • OpenExtensions authorizations • SFS authorizations.
Recovery	Establish recovery procedures.
Violation procedures	Establish security procedures for logging, reporting, auditing.
Test plan	Develop a RACF test plan.
Education	Plan to prepare user documentation and/or other educational material—perhaps a newsletter for most users, perhaps more detailed education for group administrators.
Installing RACF	Select RACF options and install RACF.
Monitoring	After beginning to define groups, users, generic profiles, and data for a pilot group, monitor progress against your implementation plan. Establish procedures to ensure that future applications receive the appropriate security considerations.

Chapter 3. Defining Users

This chapter provides in-depth information on defining users for VM systems.

As a general objective, all users should be defined to RACF. On VM, users who are not defined to RACF cannot use the system.

User Profiles

When you define a user to RACF, you create a user profile in the RACF database. A user profile can consist of a RACF segment, and optionally, an OVM segment.

Each segment of a user profile consists of fields. When you define a user's profile (using the ADDUSER command) or change a user's profile (using the ALTUSER command), you can specify the information contained in each field of each segment of the profile. You can also list the contents of an entire user profile, or the contents of individual segments of the user profile using the LISTUSER command. For information on how to use these commands, see *RACF Command Language Reference*.

The RACF Segment in User Profiles

The RACF segment of a user profile contains basic information needed to define a user to RACF. You can specify the following information in the RACF segment:

USERID	User ID of the user
NAME	User's name
OWNER	Owner of the user's profile
DFLTGRP	Default group for the user
AUTHORITY	User's authority in the default group
PASSWORD	User's password
REVOKE	Date when RACF prevents the user from gaining access to the system
RESUME	Date when RACF gives the user access to the system again
UACC	Default universal access authority for resources the user defines
WHEN	Days of the week and/or hours of the day the user can have access to the system
ADDCATEGORY	User's installation-defined security category
SECLEVEL	User's installation-defined security level
CLAUTH	Classes in which the user can define profiles
SPECIAL	Gives the user the system-wide SPECIAL attribute
AUDITOR	Gives the user responsibility for auditing system resources
OPERATIONS	Gives the user the system-wide OPERATIONS attribute
DATA	Installation-defined data
SECLABEL	User's default security label

The OVM Segment in User Profiles

When you define a new OpenExtensions VM user or change OVM attributes for an existing user, you can specify the following information in the user's profile:

UID User's OpenExtensions VM user identifier

HOME User's OpenExtensions VM initial directory path name

PROGRAM User's OpenExtensions VM program path name, such as a default shell program

FSROOT User's OpenExtensions VM file system root directory

See "Defining OpenExtensions Users" on page 241 for more information.

To define or change information in the OVM segment of a user profile, you must have the SPECIAL attribute or at least UPDATE authority to the segment by way of field-level access control. To display information in the OVM segment of a user profile, you must have the SPECIAL attribute or at least READ authority to the segment by way of field-level access control. See "Setting Up Field-Level Access for the OVM Segment" on page 247 for more information.

User Naming Conventions

The rules for naming users, like those for naming groups, are simple:

- A RACF user ID must be from 1 to 8 characters in length, and may consist of any combination of A–Z, 0–9, # (X'7B'), $ (X'5B'), or @ (X'7C')
- The characters, #, $, and @, may be displayed differently on terminals outside the United States; therefore, use the characters with the hexadecimal equivalents shown above.
- No two user IDs can be the same. No user ID can be the same as a group name.
- With OpenExtensions, the user identifier (UID) is an integer value that defines a user. Although you can use the same integer value to define multiple users, it is not recommended. If you use the same integer value:
  - Control at an individual user level is lost because the UID is used in OpenExtensions security checks.
  - Users with the same UID value are treated as a single user during OpenExtensions security checks.

  A UID of 0 is used to define an OpenExtensions superuser. A superuser can perform any OpenExtensions function and passes all OpenExtensions security checks.

Suggestions for Defining User IDs

Basically, there are no requirements for establishing a specific type of user ID. That is, in some installations, you might form user IDs by adding a numerical suffix to a group name (for example, ADMIN01, or MKT06). In other cases, you might use first names (for example, PETER and PAUL could be defined and connected to the group RESEARCH. In this case, if PETER subsequently leaves the RESEARCH group to join the TEST group, he need not change his user ID.)

The concept of user IDs based on group names appears practical because a quick glance at the user ID reveals the group. However, this concept might not prove so practical a few years later when many of the current users will have changed groups. In the long run, user IDs based on something like user names or personnel numbers do not have this problem and offer the greatest long term flexibility.

Migrating Existing User IDs to RACF

On VM, the installation procedure executes several EXECs that help in migrating users to RACF. For more information, see *RACF Program Directory for VM Installations*.

Note: The VM directory allows user IDs to have special characters such as the hyphen (-) that are not allowed in RACF. If you use the RPIDIRCT EXEC to convert VM directory statements into RACF commands, the RPIDIRCT EXEC changes the hyphens into dollar signs ($).

Creating New User IDs from Scratch

Where user IDs are being assigned from scratch, they can often be created in blocks through a user-written EXEC on VM. For example, you could centrally create 50 user IDs, MKT01 through MKT50, and allocate them to the manager of group MKT to assign to the users in the department. The default group (MKT), password, and other operands can all be preset. You should assign the REVOKE attribute to unused user IDs.

Creating New User IDs Using ISPF Panels

On VM systems, if DirMaint is installed, you can use dual registration panels to define users to both RACF and the VM directory at the same time. See “Using Dual Registration Panels” on page 296.

Ownership of a RACF User Profile

Each user defined to RACF has a user profile; all user profiles have another RACF user or group as the owner. The owner (or a user who is connected to the owning group and has the group-SPECIAL attribute, or someone with system-SPECIAL) can modify, list, and delete the user's profile and has control over the user's attributes (including the ability to prevent the user from entering the system).

For a list of the RACF commands that owners of user profiles can issue, see Table 47 on page 332.

User Attributes

User attributes are extraordinary capabilities, restrictions, or environments that can be assigned to a user either all of the time or when the user is connected to a specific group or groups. When an attribute is to apply all the time, it is specified at the system level, and is called a user attribute. When an attribute is to apply only to a specified group or groups, it is specified at the group level, and is called a group-related user attribute. For example, user attributes that you specify in an ADDUSER or ALTUSER command are indicated in the user's profile, and are in effect regardless of the group the user is connected to.

The following sections describe these user attributes:

- SPECIAL
- AUDITOR
- OPERATIONS
- CLAUTH
- REVOKE

SPECIAL Attribute

A user with the SPECIAL attribute at the system level can issue all RACF commands. The SPECIAL attribute gives the user full control over all RACF profiles in the RACF database.

The SPECIAL attribute can be delegated only by a user who has the SPECIAL attribute. It should be limited to the RACF security and group administrators. Personnel having the SPECIAL attribute should be required to use passwords, and should change their passwords often to help ensure password security.

You can assign the SPECIAL attribute at the group level. When you do, the *group-SPECIAL* user has full control over all profiles within the scope of the group. See “User Attributes at the Group Level” on page 60 for additional details.

For a list of the RACF commands that this attribute allows users to issue, see Table 41 on page 328.

AUDITOR Attribute

A user with the AUDITOR attribute at the system level has the authority to specify logging options on the:

- ALTUSER, RALTER, and SETROPTS commands on VM and MVS
- ALTDIR and ALTFILE commands on VM
- ALTDSD command on MVS.

In addition, the auditor can list auditing information with the:

- LISTGRP, LISTUSER, RLIST, and SEARCH commands on VM and MVS
- IRRUT100 utility program on VM and MVS
- LDIRECT, LFILE, SRDIR, and SRFILE commands on VM
- LISTDSD command on MVS.

The AUDITOR attribute gives the auditor control of logging to the SMF data file on VM. Logging to SMF helps to detect changes (or attempted changes) to the RACF database and accesses (or attempted accesses) of RACF-protected resources.

The user having the AUDITOR attribute can list all profile information available to the SPECIAL user, as well as information available as a result of having the AUDITOR attribute. Note, however, that this extended listing capability does not give the auditor any additional authority to change information in the RACF database; nor does it give additional access to protected data.

A user must have the AUDITOR attribute to run the DSMON program. You should assign the AUDITOR attribute only to users who are responsible for auditing RACF security controls and functions. To provide a check and balance on RACF security measures, you should give the AUDITOR attribute to security or group administrators other than those who have the SPECIAL attribute.

The AUDITOR attribute can be assigned only by a user (security or group administrator) who has the SPECIAL attribute.

You can assign the AUDITOR attribute at the group level. When you do, the *group-AUDITOR* user's authority is limited to profiles that are within the scope of that group. See “User Attributes at the Group Level” on page 60 for detailed information.

For a list of the RACF commands that this attribute allows users to issue, see Table 42 on page 329.

OPERATIONS Attribute

A user with the system-OPERATIONS attribute has full authorization to all RACF-protected resources in the DASDVOL, DATASET, DIRECTRY, FILE, GDASDVOL, PSFMPL, RODMMGR, TAPEVOL, VMBATCH, VMCMD, VMMDISK, VMNODE, and VMRDR classes, with the following exceptions:

- If users, their current connect group, or any of their connect groups (if list-of-groups checking is active) is in the access list of a resource profile, they have only the access specified in the access list. For this reason, you should plan carefully before making system-wide OPERATIONS users members of any group that is in the access lists of resource profiles.
- Security classification checking or security label checking can deny access.

You can limit the access allowed because of the OPERATIONS attribute in two ways:

- By placing the OPERATIONS user or a group the user is connected to in the access list of sensitive resources using the PERMIT, PERMDIR, or PERMFILE command. The specific access authority, such as NONE or READ, takes precedence over the OPERATIONS attribute.
- By using security levels, security categories, or security labels

Because the OPERATIONS attribute permits wide access to resources, you should assign this attribute to a minimum number of people. You should also consider auditing those users to whom you have assigned the OPERATIONS attribute. To do this, have a user with the system-AUDITOR attribute issue the following command:

```
SETROPTS OPERAUDIT
```

To reduce the number of users having the OPERATIONS attribute at the system level (and therefore having the attribute for all resources on the system, you can assign the OPERATIONS attribute at the group level. When you do, the *group-OPERATIONS* user's authority is restricted to resources within the scope of the group. For more information, see “Scope of Authority for the group-SPECIAL, group-AUDITOR, and group-OPERATIONS Users” on page 60 and “User Attributes at the Group Level” on page 60.

The OPERATIONS attribute can be delegated only by a user (security or group administrator) who has the SPECIAL attribute.

For a list of the RACF commands that the OPERATIONS attribute allows users to issue, see Table 43 on page 329.

CLAUTH (Class Authority) Attribute

Users receive the CLAUTH attribute on a *class-by-class* basis; you cannot assign the CLAUTH attribute at the user or group level. If a user has the CLAUTH attribute in a class (or in a class that shares the same POSIT value in the class descriptor table), RACF allows the user to define profiles in that class.

The classes you can specify with CLAUTH are USER and any general resource class.

Notes:

1. CLAUTH has no meaning for the FILE and DIRECTRY classes.
2. The authority of all users to define profiles in general resource classes can be restricted by issuing the SETROPTS GENERICOWNER command. See "Restricting the Creation of General Resource Profiles (GENERICOWNER Option)" on page 220.
3. A user's authority to define profiles extends to any class that has the same POSIT value in the class descriptor table (CDT). For example, if you give a user CLAUTH(TERMINAL), that user can also define profiles in class GTERMINL, because both of these classes have the same POSIT value. For the POSIT values of the IBM-supplied classes, see the description of the class descriptor table in *RACF Macros and Interfaces*.

   You should give the CLAUTH attribute only to those users who are responsible for defining profiles to RACF in the specified classes and in any classes with the same POSIT value.
4. A user to whom you assign the CLAUTH attribute for the USER class is authorized to define new users to RACF with the ADDUSER command, provided the user is the owner of or has JOIN authority in the new user's default group.

The CLAUTH attribute can be delegated only by a user with the system-SPECIAL attribute, or by a user who has both the authority to update the user profile and the CLAUTH attribute for the class authority being delegated.

For a list of the RACF commands that the CLAUTH attribute allows users to issue, see Table 44 on page 330.

REVOKE Attribute

You can prevent a RACF user from entering the system by assigning the REVOKE attribute on the ALTUSER command. This attribute is useful when you want to prevent a user from entering the system but you cannot use the DELUSER command because the user still owns RACF resource profiles.

You can also assign the REVOKE attribute on a group level by using the CONNECT command. If the user has the REVOKE attribute for a group, the user cannot enter the system by connecting to that particular group, or access resources as a member of that group.

RACF allows you to specify a future date for a REVOKE to occur (at both the system and the group level). You can also specify a future date to remove the REVOKE attribute by using the RESUME operand on the ALTUSER command.

Only the owner of a user's profile (or a user with the SPECIAL attribute) can assign the REVOKE attribute.

User Attributes at the Group Level

On both MVS and VM you can specify the SPECIAL, AUDITOR, and OPERATIONS user attributes at the group level by using the CONNECT command. When you specify these attributes at the group level, they are identified as group-SPECIAL, group-AUDITOR, and group-OPERATIONS to distinguish them from attributes at the system level.

Group attributes are indicated in the description of the user-to-group connection in the user profile. Unless list-of-group checking is active, group attributes are in effect for the user only when the user is connected to the group during a batch job or terminal session.

If list-of-groups checking is active, then, regardless of which group the user is logged on to (the current connect group), RACF recognizes the user's group-related attributes in the user's other connect groups. (It is as though the user was logged on to each group at the same time.) For more information on list-of-groups checking, see “List-of-Groups Authority Checking” on page 219.

When you initially define a new user, the user's connection to the default group does not indicate any group-related attributes. You can then use the CONNECT command to define the user's group attributes within the default group.

Scope of Authority for the group-SPECIAL, group-AUDITOR, and group-OPERATIONS Users

The authority of the group-SPECIAL, group-AUDITOR, and group-OPERATIONS users is limited to the resources that are within the scope of the group. Resources that are within the scope of the group include the following (refer to Figure 4 on page 64 and Figure 5 on page 65):

- Resources owned by the group (for example, GROUP1.DATA owned by GROUP1)
- Resources owned by users who are owned by the group (for example, USER2.DATA owned by USER2 who is owned by GROUP1)
- Resources owned by subgroups that are owned by the group (for example, GROUP2.DATA owned by GROUP2 that is owned by GROUP1)
- Resources owned by subgroups that are owned by subgroups, owned by the group, and so on (for example, GROUPZ.DATA owned by GROUPZ that is owned by GROUP2 that is owned by GROUP1).

Note that the scope of the group does *not* extend to the following resources:

- Resources owned by groups owned by users who are owned by the group (for example, GROUPY.DATA owned by GROUPY owned by USER2 who is owned by GROUP1)
- Resources owned by users who are, in turn, owned by users who are owned by the group (for example, USER6.DATA owned by USER6 who is, in turn, owned by USER5 who is owned by GROUP2).

By establishing the group structure so that subgroups are owned by their superior groups, the authority of the group-SPECIAL, group-OPERATIONS, and group-AUDITOR user can be made to percolate down through the group tree structure as far as the security administrator desires. When a user's attribute percolates down from a group to which the user is connected with the group attribute, the user's authority in the subgroups is the same as if the user was connected directly to the subgroups with the group attribute.

Note: The data security monitor (DSMON) produces a group tree report that lists, for each requested group, all of its subgroups, all the subgroups' subgroups, and so on. This report can be very useful in checking to which subgroups the authority of the group-SPECIAL, group-OPERATIONS, or group-AUDITOR applies. For more information on the group tree report, see *RACF Auditor's Guide*.

The limits of the security administrator, group administrator, auditor, and operations personnel authority at the group level are described in Table 8 on page 62. (Of course, these users continue to have whatever authorities they possess from other sources, such as ownership, that are not covered by their group-level authorities.)

Table 8. Scope of Authority for User Attributes at the Group Level

Resource	Attribute, User, and Authority
Users	**Group-SPECIAL attribute:** A user with the group-SPECIAL attribute has full authority to work with: • User profiles that are owned by the group • User profiles that are owned by a subgroup owned by the group, by a subgroup owned by a subgroup that is owned by the group, and so on. The group-SPECIAL user must have the CLAUTH attribute in a class in order to give the CLAUTH attribute to another user in that class. The group-SPECIAL user cannot give a user the SPECIAL, AUDITOR, or OPERATIONS attribute at a system level, but can assign these attributes at the group level. To create new users, the group-SPECIAL user must have the CLAUTH attribute in the USER class. **Group-AUDITOR attribute:** A user with the group-AUDITOR attribute can perform all of the functions of an auditor, but is restricted to the same subset of users as the user with the group-SPECIAL attribute.
Groups	**Group-SPECIAL attribute:** A user with the group-SPECIAL attribute has authority over that group, over subgroups owned by that group, and so on. The group-SPECIAL user can connect any user to, or remove any user from, any group that is included in this authority.
SFS files and directories	**Group-SPECIAL attribute:** A user with the group-SPECIAL attribute has full authority to work with FILE and DIRECTRY profiles: • Owned by the group • Owned by users or groups that the group owns • With a second qualifier that is a user identifier owned by the group. The group-SPECIAL user can also define FILE and DIRECTRY profiles with a second qualifier that is a user owned by the group. **Group-AUDITOR and Group-OPERATIONS attributes:** A user with the group-AUDITOR or group-OPERATIONS attribute can perform all of the functions of an auditor or operator, but is restricted to the same subset of SFS files and directories as the user with the group-SPECIAL attribute.
General resources	**Group-SPECIAL attribute:** A user with the group-SPECIAL attribute has full authority to work with: • Resource profiles that are owned by that group • Resource profiles belonging to users or groups that are owned by the group. To create new resources, the user must have the CLAUTH attribute in the applicable class. **Group-AUDITOR and Group-OPERATIONS attributes:** A user with the AUDITOR or OPERATIONS attribute can perform all of the functions of an auditor or operator, but is restricted to the same above subset of resources as the user with the group-SPECIAL attribute.

The following two figures show the scope of authority of a group-SPECIAL user. Figure 4 on page 64 shows a typical authority structure containing three major groups: Group 1, Group 2, and Group 3.

Note: Although this figure uses MVS terminology to illustrate relationships between users and groups and their resources, the same concepts apply to relationships between users and groups and their resources on VM systems.

Figure 5 on page 65 shows the addition of a new element: a new user, USER1, is connected to Group 1. The resultant authority USER1 receives as a group-SPECIAL user is highlighted (the nonshaded area) in part 2 of this figure.

USER1 has authority to the profiles in the nonshaded area for the reasons listed in Table 8 on page 62. USER1 does *not* have authority to any of the resources in the shaded area for the following reasons:

- GROUP1 does not own IBMUSER, GROUP3, USER3, or USER4.
- GROUP1 does not own GROUPY.
- Neither GROUP1 nor GROUP2 own USER6.
- USER3.DATA is not owned by a user who is owned by GROUP1.
- USER4.DATA is not owned by a user who is owned by GROUP1. If USER4.DATA is an MVS data set, USER1 cannot display the profile information for this data set with LISTDSD, even if USER2, for example, is in its access list. (However, by using IRRUT100, RACF informs USER1 that USER2 is in the access list of USER4.DATA.)
- U4A is not a general resource owned by a user who is owned by GROUP1.

Note: Although this figure uses MVS terminology to illustrate relationships between users and groups and their resources, the same concepts apply to relationships between users and groups and their resources on VM systems.

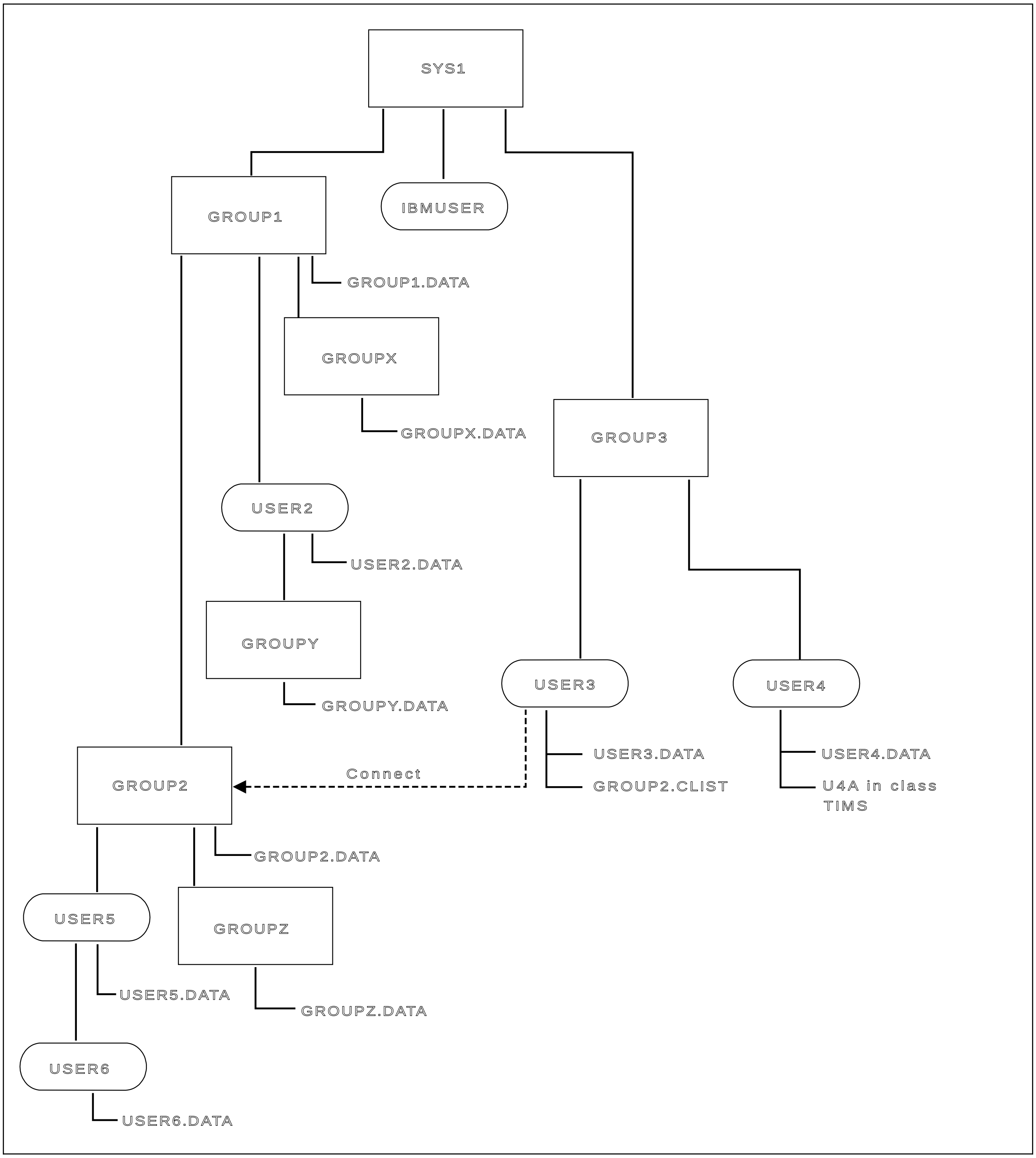


Figure 4. Group-Level Authority Structure

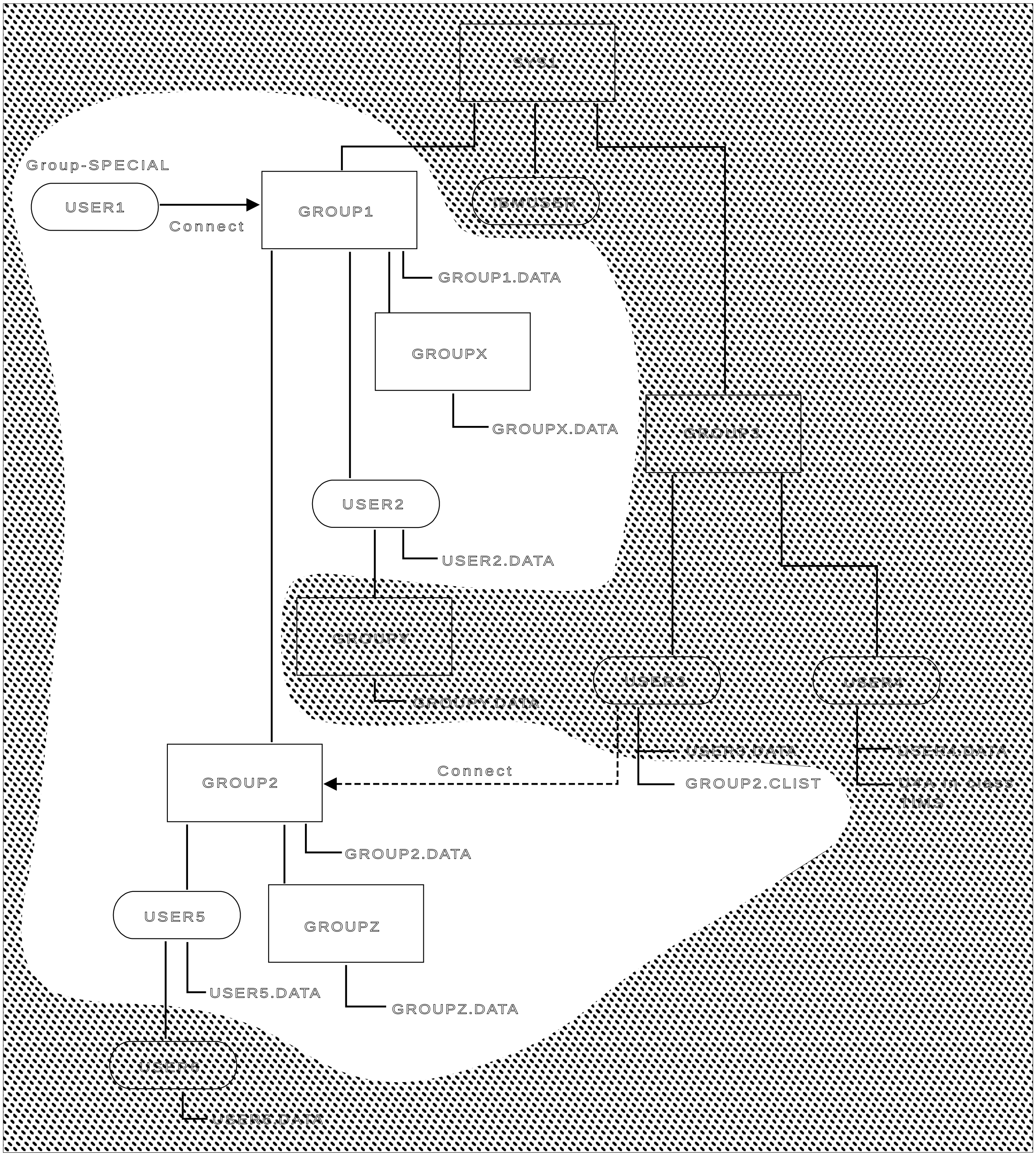


Figure 5. Scope of Authority of a Group-SPECIAL User

Suggestions for Assigning User Attributes

When defining users to RACF with the ADDUSER command, or when modifying user attributes with the ALTUSER command, RACF security and group administrators should assign:

- SPECIAL, AUDITOR, and OPERATIONS attributes to only those users responsible for administering RACF on a system-wide basis
- CLAUTH attributes to only those users who will define other users and general resources (other than SFS files and directories).

Verifying User Attributes

The data security monitor (DSMON) generates reports that describe the current status of the data security environment at your installation. Two of these reports, the selected user attribute report and the selected user attribute summary report, are useful for verifying the attributes that you have assigned.

The selected user attribute report lists all RACF users with the SPECIAL, OPERATIONS, AUDITOR, or REVOKE attributes and specifies whether they possess these attributes on a system-wide (user) or group level. You can use this report to verify that only those users whom you want authorized to perform certain functions have been assigned the corresponding attribute.

The selected user attribute summary report shows the number of installation-defined users and totals for users with the SPECIAL, OPERATIONS, AUDITOR, and REVOKE attributes, both at the system and group level. You can use this report to verify that the number of users with each of these attributes, on either a system or group level, is the number that your installation wants.

Default Universal Access Authority (UACC)

Each user connected to a group is assigned a default universal access authority (UACC) of NONE, READ, UPDATE, CONTROL, or ALTER. You can specify this default UACC on the ADDUSER, ALTUSER, or CONNECT command. If you do not specify a value for UACC, RACF uses NONE as a user's default universal access authority.

RACF uses this default UACC for all new resources a user defines while connected to the specified default group. When a user issues the ADDDIR, ADDFILE, or RDEFINE command to define a new general resource profile and does not specify a value for the UACC operand, RACF uses the default UACC as the UACC for the profile unless a value for UACC is specified in the class descriptor table.

For more information on using the UACC operand on the ADDUSER, ALTUSER, or CONNECT command, see *RACF Command Language Reference*.

Assigning Security Categories, Labels, and Levels to Users

On MVS and VM, you can assign security categories and security levels to users and sensitive resources. You can also assign security labels, which are a combination of security levels and security categories, and are more easily maintained, to users and sensitive resources. User profiles and resource profiles that have been assigned a security label need not be changed if the definition of a security label is changed.

A security category is an installation-defined name corresponding to a department or area within an organization that has similar security requirements; a security level is an installation-defined name that is associated with a number in the range 1 through 254.

If security levels and categories are being used (the SECDATA class is active), and security labels are *not* being used (the SECLABEL class is not active), RACF takes the following steps when a user requests access to a resource that has a security category or a security level associated with it:

1. If the resource has a SECLEVEL, RACF compares the security level of the user with the security level of the resource. If the resource has a higher security level than the user, RACF denies the request. For a terminal session, the security level that RACF uses for the user is the lower of the user's SECLEVEL and the terminal's SECLEVEL. Thus if the terminal has a SECLEVEL of 50 and the user has a SECLEVEL of 100, the user cannot access, through that terminal, any data that has a SECLEVEL of over 50. RACF then proceeds to the category check.

   If the resource does not have a SECLEVEL, then RACF proceeds to the category check in Step 2.

2. RACF compares the list of security categories in the user's profile with the security categories in the resource profile. If the user's security level is high enough to access the resource, RACF compares the list of security categories in the user's profile with the list of security categories in the resource's profile. If RACF finds any security category in the resource profile that is not in the user's profile, RACF denies the request. If RACF does not deny the request, RACF continues with authorization processing. If there are no categories in the resource profile, RACF continues with authorization processing.

If your installation has activated the SECLABEL class, and a user requests access to a resource that has a security label associated with it, RACF *ignores* any security level or security categories specified in the resource profile. Instead, RACF performs security label authorization checking, which involves the security levels and categories used to define the security labels of the resource and the user.

You can use security labels as a simple replacement for security levels and categories, with the same access authority requirements, or you can use SETROPTS options such as MLACTIVE and MLS to set up a more rigorous security environment. (For more information on how the SETROPTS options change the effects of security labels, see “Security Label Authorization Checking” on page 315.) See also Page 11.

For more information on setting up and using security classification of users and data, see Chapter 7, “Security Classification of Users and Data” on page 111.

Limiting When a User Can Access the System

Installations can restrict a user's ability to log on by limiting:

- The user's ability to log on to the system to certain days of the week, and certain hours within each day
- The use of individual terminals (in the TERMINAL class only) to certain days of the week, and certain hours within each day.

To restrict a user from entering the system, use the WHEN operand on the ADDUSER and ALTUSER commands. For example:

```
ADDUSER USER12 WHEN(DAYS(WEEKDAYS) TIME(0700:1700))
```

specifies that USER12 can enter the system only on weekdays between the hours of 7:00 a.m. and 5:00 p.m.

Similarly, in order to control when users can access the system from a specific terminal, specify the WHEN operand on the RDEFINE and RALTER commands for the appropriate profile. For example, for a profile in the TERMINAL class:

```
RDEFINE TERMINAL TRM07C WHEN(DAYS(WEEKDAYS))
```

specifies that terminal TRM07C can be used at any time during the week, but not at all during the weekend. (When specifying the RDEFINE command, TIME(ANYTIME) is the default.)

The WHEN operand on these commands (for both users and terminals) allows you to specify individual days and specific times within these days.

RACF also provides support for installations that have terminals in different time zones by associating with each terminal its location relative to the local time where the processor complex on which RACF is executing is located.

Note: RACF does not provide any specific support for daylight savings time. If the installation changes the value of the local time (as given by the TIME macro instruction) to accommodate daylight savings time, RACF automatically adjusts its time calculations accordingly. However, if any terminals are located in an area that does not follow the same time adjustment, you must adjust the terminal information.

For more information on the WHEN operand, see the command descriptions in *RACF Command Language Reference*.

User or Terminal Time and/or Day-of-Week Checking

The time and day-of-week checking for users and terminals applies when users log on to terminals from VM. User verification processing includes the following time and day-of-week checks:

1. The user's authority to use the specific terminal. If the profile protecting the terminal does not have any time or day-of-week information, the user can log on. If there is time and day-of-week information, RACF calculates the time-of-day at the location of the terminal the user is logging on from (if that time is different from the time-of-day at the location of the processor complex), and checks whether the terminal can be used at this time on this day of the week.

2. The user's authority to access the system. If the user's profile does not have any time or day-of-week information, the user can log on. If there is time and day-of-week information, RACF calculates the time-of-day at the location of the terminal the user is logging on from (if that time is different from the time-of-day at the location of the processor complex), and checks whether the user can enter the system.

Defining Shared User IDs

Using the RACF LOGON BY function, you can define shared user IDs. Multiple users can log on to another user ID using their own password. This function uses:

1. The BY option of the LOGON command to specify the surrogate user
2. The SURROGAT class to perform authorization checks for logons to shared user IDs

RACF allows one user ID to log on to a shared user ID if that user has at least READ access to the SURROGAT profile named LOGONBY.*shared_userid*.

To understand the function, you need to become familiar with the following terms:

shared user ID User ID that has the capability of being logged onto by a different user.

surrogate user Person logging on to the shared user ID.

direct logon A “traditional” logon, in which you log on to your own user ID.

shared logon A logon in which a surrogate user uses the BY option of the LOGON command to logon to a different user ID. The surrogate user operates with the RACF authority of the shared user ID.

With this support, you can share user IDs without compromising system security. To protect system security:

- The surrogate user's password is used to access the shared user ID. This avoids the need for users to share passwords.
- An audit trail identifies the surrogate user whenever work is performed by the shared user ID.

Interaction between RACF and VM

VM provides the LOGONBY directory statement to authorize shared LOGONs when no external security manager is installed. A description of how this statement is used when RACF is installed follows. For details on VM's LOGONBY directory statement, see *z/VM CP Planning and Administration*.

When RACF Is Active

When RACF is active, VM ignores the LOGONBY directory statement and leaves the access decision to RACF. RACF determines authorization based on the SURROGAT class profiles.

When RACF Is Inactive

When RACF has been set inactive by the SETRACF INACTIVE command, it defers the shared LOGON authorization decision to VM.

- If the surrogate user ID is specified on the shared user ID's LOGONBY directory statement, VM checks the password supplied by the surrogate user against the password in the surrogate user's directory entry. If these checks succeed, the shared LOGON is allowed.
- If the user is authorized in the VM directory but not in the appropriate RACF SURROGAT profile, the shared logon succeeds when RACF is inactive. Once RACF is active again, it audits the user as a normal shared user ID for the duration of the logon.

When the RACF Service Machine Is Uninitialized or Unresponsive

When the RACF service machine is uninitialized or unresponsive, no users are allowed to log on to the system, except for the following:

- Any RACF service machine
- The primary system operator.

For these user IDs, the same considerations apply as when RACF is inactive. See “When RACF Is Inactive.”

Note: By coding LOGONBY directory statements for the primary system operator or for RACF service machines, you still allow the use of LOGON BY for these user IDs when the RACF service machine is experiencing problems. This could be useful for recovery purposes.

Setting Up the LOGON BY Function

To let users share user IDs, perform the following tasks:

Step 1 Define profiles of the form LOGONBY.*shared_userid* in the SURROGAT class for each user ID that is to be shared.

Step 2 Permit specific users to the appropriate SURROGAT profiles.

Step 3 Activate the SURROGAT class.

RACF checks for a SURROGAT profile on *all* logon attempts, both shared and direct, To improve performance, you may want to RACLIST the SURROGAT class. To do this, enter:

```
SETROPTS RACLIST(SURROGAT)
```

If the SURROGAT class is not active or the LOGONBY.*shared_userid* profile is not defined, the shared logon fails with:

```
  RPIMGR065A Surrogate user authorization failed for user ID userid
```

Allowing Access to a Shared User ID

You can let a user ID log on to a shared user ID by defining the SURROGAT profile and permitting that user ID to that SURROGAT profile. For example, to allow BRUCEW to log on to the shared user ID DIRMAINT, the administrator can permit BRUCEW to the correct SURROGAT profile as follows:

```
   RDEFINE SURROGAT LOGONBY.DIRMAINT UACC(NONE)
   PERMIT LOGONBY.DIRMAINT CLASS(SURROGAT) ID(BRUCEW) ACCESS(READ)
```

Direct and Shared Logon

You can allow users to log on to a shared user ID directly or as shared.

Logging On as Shared: By default, a user ID that is defined as shared may not be logged on to directly.

For example, a service machine such as DIRMAINT needs an audit trail to identify which user has logged on to it. Therefore, users should not be allowed to log on directly.

However, because the administrator permitted BRUCEW to the LOGONBY.DIRMAINT profile in the SURROGAT class, BRUCEW can log on as shared to the user ID DIRMAINT with the BY option of the LOGON command.

Logging On Directly: You have the option of letting users log on to a shared user ID directly. This may be desirable in some cases.

For example, manager DONNA wants her secretary BRADPITT to log on to her user ID, but also needs to log on to her own user ID. You can authorize this by defining and permitting the manager to the SURROGAT profile as follows:

```
RDEFINE SURROGAT LOGONBY.DONNA UACC(NONE)
PERMIT LOGONBY.DONNA CLASS(SURROGAT) ID(DONNA)    ACCESS(READ)
PERMIT LOGONBY.DONNA CLASS(SURROGAT) ID(BRADPITT) ACCESS(READ)
```

In this case, DONNA can log on to her own user ID whenever she needs to use it.

Special Considerations

You need to consider the following when using the LOGON BY function.

Ownership Considerations

If the shared user ID belongs to a person who wants to let other people log on to that ID, you can make that person's user ID the owner of the SURROGAT profile. The owner can permit people to log on to the user ID without requiring intervention from an administrator.

For example, you can make DONNA the owner of her SURROGAT profile as follows:

```
RALTER SURROGAT LOGONBY.DONNA OWNER(DONNA)
```

You should be aware that when a surrogate user logs on to that user ID, the surrogate user has the authority to permit other users to log on to the shared user ID.

Authorization Considerations

If you change the surrogate user's authorization to a shared user ID's SURROGAT profile while that shared user is logged on, the change does not take effect during the current logon session. It takes effect the next time the shared user ID logs on.

For example, while BRUCEW is logged on to DIRMAINT, the security administrator changes his access from READ to NONE in the SURROGAT profile. BRUCEW can use DIRMAINT for the remainder of the session, but cannot log on to the ID again.

Terminal Considerations

If the TERMINAL class is active when a surrogate user tries to log on to a shared user ID, both the shared user ID and the surrogate user must have access to the terminal being used.

Security Label Considerations

If the SECLABEL class is active, both the shared user and the surrogate user must be permitted to the appropriate SECLABEL profile. RACF uses the SECLABEL from the first place it is found subject to the following order:

1. The SECLABEL specified on the LOGON command
2. The SECLABEL specified as the surrogate user's default SECLABEL in its USER profile
3. The SECLABEL specified as the shared user ID's default SECLABEL in its USER profile

If no SECLABEL is found in any of these places, the SETROPTS MLACTIVE setting determines whether the user can logon without a SECLABEL. If MLACTIVE(FAIL) is in effect, a user cannot log on without a SECLABEL.

If the SECLABEL class is active and a SECLABEL exists in the SURROGAT profile, but a SECLABEL was neither specified on the command line nor in the surrogate user's USER profile, the shared logon attempt fails. In this case, the user logging on receives message RPIMGR065A and the security console receives message ICH408I indicating that the submitter is not authorized by the user.

General Considerations for User ID Delegation

- In general, centralize first, delegate later.
- Consider the trade-offs:
  - Should one user handle all of the administration workload?
  - Should many users all be learning RACF simultaneously?
- RACF groups (not users) should own resource profiles.
- Authorize groups to resource profiles rather than users.
- Delegate power (group-SPECIAL) with care.
- Have “standby” SPECIAL and OPERATIONS user IDs for emergency situations.

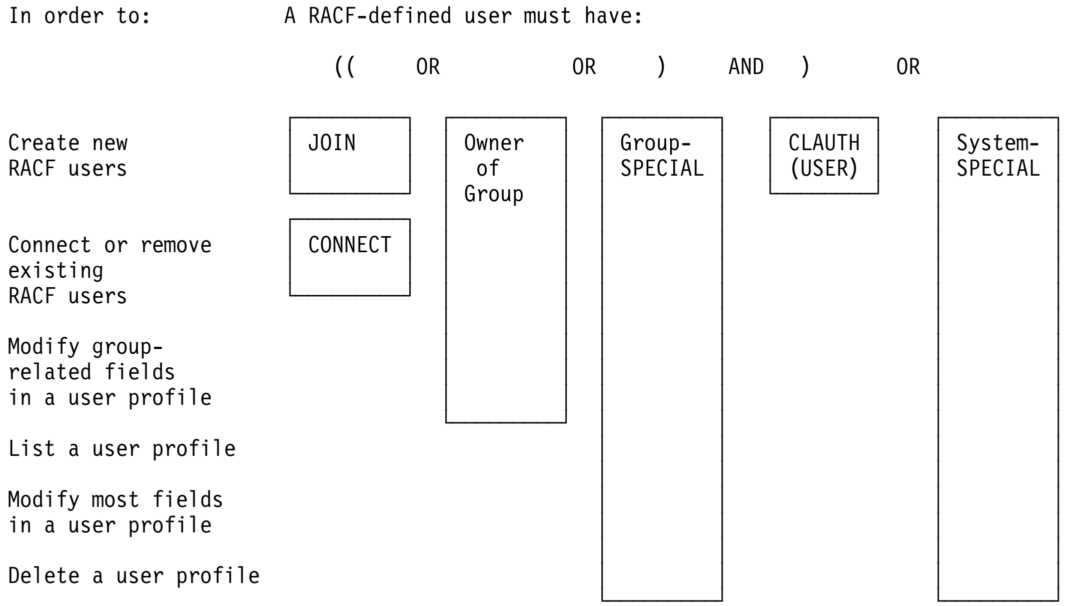
  Note: The password for the “standby” user IDs should be kept under lock and key.
- Once control has been given, it is difficult to take it away again.
- Group-SPECIAL is the most powerful authority a user can have at the group level.
  - Group-SPECIAL enables the user to use more commands.
  - Group-SPECIAL also percolates to other groups, as far as the scope of the group allows.

Choose the best *current* option for your installation.

- For authority over a *single* group of resources based on protection objectives, use JOIN and CLAUTH(USER).
- For authority over one or more groups of resources based on protection objectives and scope of the group, use group-SPECIAL and CLAUTH(USER).

Note: The group-SPECIAL attribute allows password resetting for user IDs within the group while JOIN does not.

Figure 6 shows delegating authority in another way.



A combination of authorities is required to create new RACF users.

For example, to create a new RACF user, the creating user must have at least one of the following *and* have the CLAUTH(USER) attribute:

- JOIN group authority in the new user's default group

  or

- Be the owner of the new user's default group

  or

- Have group-SPECIAL in the new user's default group

or

- Have system-SPECIAL.

Figure 6. Delegating Authority (User Profiles)

Summary of Steps for Defining Users on VM

This summary presents the steps required by RACF and related IBM licensed programs to define users to RACF on VM. Your installation may require additional steps, depending on your security policy and the products you have installed.

1. Prepare to create the user profile as follows:

   - Decide which default connect group to assign to the user. If a group profile does not yet exist for the group, create the group. See “Summary of Steps for Defining a RACF Group” on page 85.
   - Decide which user ID to assign to the user.

     Notes:

     a. This must match the user's directory entry.

     b. If your installation is using access control groups, (ACIGROUP control statement in directory entries), see “Use of ACIGROUP Control Statements” on page 194.
   - Decide which user or group is to be the owner of the user profile. (If the owner is a user, provide him or her with the necessary information for managing the new profile.)
   - Decide what initial password is to be assigned to the user. (If you do not specify a password, the new user's default group name becomes the new user's initial password. You might prefer to specify a non-trivial password.)
   - Determine if the user's access to the system should be restricted to certain days of the week, hours of the day, or both.
   - Decide which user attributes (such as SPECIAL or AUDITOR) the user should have, and whether the user attributes should be limited to the scope of a group (group-SPECIAL or group-AUDITOR).
   - If security labels are used, decide which default security label to assign to the user.

2. Create the user profile.

   Note: This can be done using any of the following methods:

   - Issuing the ADDUSER command.
   - If DirMaint is installed, using the dual registration panels supplied with RACF for VM.

3. Create minidisk profiles for each of the user's minidisks.

   Note: This can be done using one of the following methods:

   - Using the RDEFINE command:

     ```
     RDEFINE VMMDISK LWARD.191 UACC(NONE)
     ```
   - Using the dual registration panels.

   Note: Discrete profiles are recommended for the VMMDISK class. For more information, see “Protecting VM Minidisks” on page 165.

4. If the user will be enrolled in an SFS file pool, create the appropriate generic profiles. For example, if ANDREW will be enrolled in file pool POOL1, enter:

```
ADDDIR POOL1:ANDREW.** OWNER(ANDREW)
ADDFILE * * POOL1:ANDREW.** OWNER(ANDREW)
```

5. If users at your installation manage their own resource profiles, provide the user with the necessary information for doing so. For example, the user might need to use portions of the *RACF General User's Guide*.

6. If the user is to define general resource profiles (as, for example, an administrator might), give the user the CLAUTH attribute in the appropriate classes, and provide the user with information for working with those profiles. For example:

   ```
   TAPEVOL class
   TERMINAL class
   VMSEGMT class
   ```

7. If needed, give the user access to RACF-protected resources. This can be done using one or both of the following:

   - Connect the user to groups that have the same access requirements as this user.

     Note: This can be done using the CONNECT command.

   - If the user requires specific access to RACF-protected resources (beyond that permitted by connecting the user to groups), give the user the access required.

     Note: This can be done using the PERMIT command.

Deleting a User

Summary of Steps for Deleting Users on VM

This summary presents the steps required by RACF and related IBM program products to delete users from RACF on VM. Your installation may require additional steps, depending on your security policy and the products you have installed.

1. To prevent the user from entering the system, revoke the user ID:

   ```
   ALTUSER  userid  REVOKE
   ```

2. Check one of the following two sections to delete the user profile and all occurrences of the user ID from the RACF database:

   - “Running the RACFDEL and RPIDELU EXECs”
   - “Deleting a User Manually” on page 76.

Running the RACFDEL and RPIDELU EXECs

RACFDEL and RPIDELU are sample EXECs that are shipped on the RACF product tape.

RACFDEL uses the work file produced by the RACUT100 EXEC (IRRUT100 SYSUT1) as input, and produces RACF commands in a file called RDEL CMDS as output.

Requirements for using RACFDEL are:

- You must have the system-SPECIAL attribute.

- RACFDEL cannot be run from either the primary or the backup RACF service machine.
- Make sure that you have READ access to the file containing current output from the RACUT100 EXEC (the file is IRRUT100 SYSUT1).

RACFDEL prompts for:

- The user ID to be deleted.
- The user ID or group ID of a new owner for resources owned by the user being deleted. You can supply a single user ID to be the new owner of all the old user's resources, or you can supply a different user ID for each resource that needs a new owner.

RPIDELU executes the commands that RACFDEL produces. Like RACFDEL, you must have the system-SPECIAL attribute to use RPIDELU. Before you execute RPIDELU, *examine the RACF commands in RDEL CMDS carefully*. Some of the things you should look for are:

- Data set commands

  If you are sharing the RACF database with MVS, RDEL CMDS may contain some data set commands. In this case, you should transmit the file to MVS for execution under the TMP. You should remove the NOSET operand from DELDSD commands before the job stream is executed on MVS. If you execute the job stream on VM, the NOSET operand must not be removed.

- REMOVE from default group

  Because RACFDEL cannot tell which group is the user's default group, RDEL CMDS will contain a REMOVE command to remove the user from his or her default group. When this command is executed, RACF will produce an error message. You can avoid this message (which can be ignored) by deleting the command that attempts to remove the user from his or her default group.

- RDELETE for SURROGAT, VMBATCH, and VMRDR profiles

  RDEL CMDS contains RDELETE commands to delete the profiles for the user's minidisks and SFS files and directories.

  RDEL CMDS does *not* contain commands to delete LOGONBY.*userid* (SURROGAT), batch (VMBATCH), or reader (VMRDR) profiles. If you want to delete profiles for these resources from the RACF database, add the appropriate RDELETE commands to RDEL CMDS.

See step 5 under “Deleting a User Manually” for information about deleting BFS files and EXEC.U*uid* and EXEC.G*gid* profiles.

Deleting a User Manually

You can delete a user's user profile manually and remove all occurrences of the user ID from the RACF database.

1. If the user owns any resource profiles (the user's user ID appears in a qualifier of the profile name), delete the resource, or, for minidisks, transfer the minidisk to another user.

Notes:

a. You can use the dual registration panels to transfer a minidisk to another user.

b. You can use the following SEARCH command to identify which general resource profiles includes the user ID in the profile name.

```
SEARCH  CLASS(class-name)  FILTER(**.userid.**)
```

If you use the RAC command processor, the output of the SEARCH command is written to a file named RACF DATA. You can then edit this file to generate RDELETE commands to delete profiles. You should consider using this SEARCH command in any class with profile names that can include user IDs: FILE, DIRECTRY, SURROGAT, VMBATCH, VMCMD, VMMDISK, and VMRDR.

2. Work with the VM administrator to delete the user's entry from the VM directory.

   Note: If you use the dual registration panels to delete the user, the user's VM directory entry is also deleted.

3. Remove the user from any access lists in which the user's user ID is specified.

   Notes:

   a. To research this step, use the IRRUT100 utility to list the occurrences of the user ID in the RACF database.

   b. To do this, use the DELETE operand on the PERMIT command.

4. If the user owns any RACF profiles, change the OWNER field of the profile.

   Notes:

   a. To research this step, use the IRRUT100 utility to list the occurrences of the user ID in the RACF database.

   b. To do this, use the appropriate RACF alter command, such as ALTUSER or RALTER.

5. If the user ID has a UID assigned to it in the OVM segment of its USER profile, check for any files in the BFS that are owned by the UID. These files will either need to be deleted or transferred to another UID using the **chown** command. For more information about **chown**, see *OpenExtensions for z/VM: Command Reference.*

   If the UID owns any set-UID or set-GID executable files, there may be profiles in the VMPOSIX class that need to be deleted or renamed appropriately. These profiles have the format:

   ```
   EXEC.Uuid
   EXEC.Ggid
   ```

   See “Protecting Set-UID and Set-GID Executable Files” on page 253 for details on these profiles.

6. After all occurrences of the user ID are deleted from the RACF database, delete the user profile.

   Note: This can be done using any of the following methods:

   - Issuing the DELUSER command
   - Using the dual registration panels.

Chapter 4. Defining Groups

This chapter provides in-depth information on defining groups for VM systems.

The group structure of RACF can be mapped into the organizational structure that exists at your installation. That is, RACF conforms naturally to a tree structure of groups, with each group except the highest (the IBM-supplied SYS1 group) having a superior, or owning, group. Groups can correspond directly to business entities such as divisions, departments, and projects; users can be connected to one or more groups.

When you define a group, you should consider the basic purpose of the group. That is, is it an administrative group, a holding group, a functional group, or a user group?

When setting up RACF groups, you might want to keep in mind that the maximum number of users that you can connect to any one group is approximately 5900. See *RACF System Programmer's Guide* for information about how to determine the exact maximum number.

Administrative Group: You can create a group simply as an administrative convenience. For example, you might create a group to represent an organizational entity, such as a region or a division.

With RACF delegation, you could create this kind of group for each group administrator. Operating from such groups, the group administrators could then define other groups needed by their local users.

Holding Group: A popular technique that retains user definition centrally, yet allows effective use of group administrators, is to establish a *holding group*. You define all users centrally and initially connect them to a group named HOLD with the minimum of authorities. HOLD does not appear in any access lists, and therefore is of no real significance to the user.

Group administrators, to whom you then give CONNECT (but not JOIN) authority, connect the appropriate users to the groups under their control and change the users' default group name as appropriate. This technique allows the installation to assign correct account numbers and control other installation considerations while allowing flexibility in the grouping of the user population.

Note: A group cannot contain more than approximately 5900 users. Therefore, if you have more than this number of users, you cannot assign them to a single holding group. Also, you should be aware that extremely large groups can have performance implications for the RACF database. For more information, see *RACF System Programmer's Guide*.

Functional Group: A group can represent a functional area of the installation for the purpose of data sharing. For example, a financial analyst might need to access a variety of resources across many groups, such as accounting, payroll, marketing, and others. Of course, the owners of each resource could permit the financial analyst to access their resources by placing the analyst's user ID on an access list. But if a new financial analyst takes over the job, it is then necessary to add the new user ID to each RACF profile. Likewise, the RACF profiles will have to be updated when the analyst no longer has a need to access the data. This arrangement involves a great deal of unnecessary activity by the resource owners.

Instead, you can create a group that represents the financial analyst function and permits access to the data defined to the group. Access to the entire range of data can then be managed by controlling the user population in the defined group. For those cases involving one-time access, owners of the needed data would simply PERMIT access by the defined group. Where appropriate, the group name could be included in profile access lists to ensure automatic availability of needed data to the financial analyst group. New financial analysts could be connected to the group, as required, to gain access to the entire range of data. Likewise, analysts could be removed from the group whenever necessary. By controlling the user population of such a functional group, resource profile changes on a day-to-day basis become unnecessary.

User Group: You can define a group to serve as an anchor point for users who otherwise have no common access requirements. For example, engineers and scientists, as well as other problem-solving users, might have no need to access application-related data in the system. Their only interest might be in their own personal data. You can place this set of users in a single group that has no access to other data.

Also, you can define groups based on access level. For example, if PAY.195 is a VM minidisk, two groups could be defined, PAYREAD and PAYUPDTE, both of which would appear in the PAY.195 access list, but with READ and UPDATE access, respectively. Any users requiring access would be connected, as appropriate, by the group administrator.

Group Profiles

When you define a group to RACF, you create a group profile in the RACF database. A group profile can consist of the following parts, or *segments*: a RACF segment, and optionally, an OVM segment.

Each segment of a group profile consists of *fields*. When you define a group's profile (using the ADDGROUP command) or change a group's profile (using the ALTGROUP command), you can specify the information contained in each field of each segment. You can also list the contents of an entire group profile, or the contents of individual segments of the group profile using the LISTGRP command. For information on how to use these commands, see *RACF Command Language Reference*.

The RACF Segment in Group Profiles

The RACF segment of a group profile contains basic information needed to define a group to RACF. You can specify the following fields in the RACF segment:

group-name Name of the group

OWNER Owner of the group's profile

SUPGROUP Name of the superior group

TERMUACC/NOTERMUACC Indicates whether to allow access based on the UACC of the terminal profile (for terminals protected by RACF)

DATA Installation-defined data

The OVM Segment in Group Profiles

You can use the OVM segment of the group profile to specify information about the group's OpenExtensions VM group. Specifically, when you define a new OpenExtensions VM group or change OVM attributes for an existing group, you can specify the following information in the group's profile:

GID Specifies the group's OpenExtensions VM group identifier

When a user is connected to a group that has a GID value assigned to it, that GID can be used in determining a user's access to files in the byte file system (BFS), in accordance with the rules of file access defined by the POSIX standard. See "Defining OpenExtensions Groups" on page 245 for more information.

To define or change information in the OVM segment of a group profile, you must have the SPECIAL attribute or at least UPDATE authority to the segment by way of field-level access control. To display information in the OVM segment of a group profile, you must have the SPECIAL attribute or at least READ authority to the segment by way of field-level access control. For more information, see "Field-Level Access Checking" on page 209.

Group Naming Conventions

The group naming conventions are relatively simple:

- A group name must be from 1 to 8 characters long, chosen from the letters (A–Z), numbers (0–9), or # (`X'7B'`), $ (`X'5B'`), or @ (`X'7C'`). It may not start with a number.

  Note: These characters may be displayed differently on terminals outside the United States; therefore, use the characters with the hexadecimal equivalents shown above.

- No two groups can have the same name. No group name can be the same as a user ID.

Since there is a potential that two or more users might want to use the same group name (for example, ADMIN), you should adopt naming standards locally to prevent this. Consider, for example, assigning a unique 1- or 2-character group name prefix to each group administrator. Then each group defined by a group administrator would have a name consisting of the administrator's prefix followed by whatever characters the administrator chooses to use. This prefixing ensures that two group administrators cannot use the same group name.

With OpenExtensions, the group identifier (GID) is an integer value that defines a group. Although the same value can be used to define multiple groups, it is not recommended. If you use the same value, control at an individual group level is lost because the GID is used in OpenExtensions security checks. Groups with the same GID value would be treated as a single group during OpenExtensions security checks. See “Defining OpenExtensions Groups” on page 245 for more information.

Benefits of Using RACF Groups

This section describes some of the ways that RACF groups can be particularly useful.

Reducing the Effort of Maintaining Access Lists

Instead of adding and deleting users to the access lists of several profiles, consider creating a RACF group and placing it on the access list instead of the user IDs. Then, give CONNECT group authority in that group to an appropriate person (perhaps the owner of the resource profiles). That person can then change the membership of the group (through the use of the CONNECT and REMOVE commands) instead of issuing the PERMIT command many times to change the access lists of all the affected resource profiles.

Avoiding the Need to Refresh In-Storage Profiles

If your installation maintains in-storage copies of resource profiles through the SETROPTS RACLIST or SETROPTS GENLIST command, changes to those profiles do not take effect on the system until a SETROPTS RACLIST REFRESH or SETROPTS GENERIC REFRESH command is issued.

To avoid the need to refresh the in-storage copies, place a *RACF group* on the access list instead of a user ID. Then, give CONNECT group authority to an appropriate person (perhaps the owner of the resource profiles). That person can then change the membership of the group (through the use of the CONNECT and REMOVE commands) instead of issuing the PERMIT command to change the access list of the affected resource profiles, and asking a user with system-SPECIAL to refresh the in-storage profiles.

Providing a Form of Timed PERMIT

You can allow a user to access a protected resource for a limited time by taking the following steps:

1. Ensure that the only access the user has to the resource is by virtue of the fact that the user is connected to a RACF group that has the desired access to the resource. (List the appropriate resource profile(s) to check for the user's user ID, or other groups to which the user is connected, in the access list. Also, list the user's RACF user profile to check for the system- or group-OPERATIONS

attribute. Depending on the class of the resource, having the OPERATIONS attribute might allow the user to access the resource.)

2. Connect the user to the group with a resume or revoke date. To cause the user's access to stop on a certain date, issue the following command:

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

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

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

   Attention

   If the user's membership in the group allows the user to create profiles, and the user becomes the OWNER of such profiles, the user might still have access to the profiles after the revoke date.

Group Ownership and Levels of Group Authority

The following topics describe the various aspects of group ownership, group authorities, suggestions for assigning group authorities, and the group terminal option.

Ownership of a RACF Group

Each group you define to RACF must be owned by a RACF-defined user or by its superior group. You assign ownership of a group with the ADDGROUP command when creating a new group profile, or with the ALTGROUP command when altering an existing group profile. If you are the owner of a group (or if you are a connected user having the group-SPECIAL attribute), you have the authority to:

- Define new users to RACF (provided you also have the CLAUTH attribute for the USER class)
- Connect and remove users from the group
- Delegate and change group authorities and set the default UACC for all new resources belonging to members of the group
- Modify, list, and delete the group profile
- Define, delete, and list the names of the subgroups under the group
- Specify the group terminal option.

Note: Ownership of a group by a user does not give that user the ability to update the access lists of resource profiles owned by the group.

For a list of the RACF commands that group owners can issue, see Table 47 on page 332.

Group Ownership of Profiles

You can assign a *RACF group* as the owner of a user profile, group profile, data set profile, minidisk profile, or any other general resource profile. In this way, profile ownership can remain constant, regardless of how often users change jobs in your organization.

Any user connected to the owning group who has the group-SPECIAL attribute will have the authority of SPECIAL for all profiles owned by the group (see “User

Attributes” on page 56) and will have the ability to perform all owner functions for the group.

You can assign any group to be the owner of a profile. (A group profile must be owned by a user or by its superior group.) An owning group does not need to be a group to which a user (represented by the profile) is connected. Being able to assign any group as an owner allows you flexibility in defining an authority structure. For example, you could establish one group for the sole purpose of owning user profiles, and give a group administrator the group-SPECIAL and CLAUTH (for the USER class) attributes in that group.

Group Authorities

Each user in a group requires a level of group authority for that group. If a user is connected to several groups, the user has a level of group authority for each group. The various group authorities are described in Table 9.

Table 9. Group Authorities

Authority	Functions Permitted	RACF Commands Permitted
USE	A user with the USE group authority can enter the system under control of that group and get access to resources (such as minidisks, terminals, and SFS files and directories) the group is authorized to.	LDIRECT LFILE LISTDSD RLIST
CREATE	On MVS, a user with the CREATE group authority can allocate new group data sets, RACF-protect group data sets, and control access to the profiles he or she has created. However, the user cannot delete the data sets, unless the user has access other than the CREATE group authority itself. CREATE group authority includes the privileges of USE group authority.	ADDSD command for group data set profiles (all operands except NOSET)
CONNECT	A user with CONNECT group authority can connect users (who are already defined to RACF) to the group and assign USE, CREATE, or CONNECT group authority to users in the group. CONNECT group authority includes the privileges of USE and CREATE group authorities.	All of the above, plus: • ALTUSER (only GROUP, AUTHORITY, or UACC operands) • CONNECT (all operands except SPECIAL, NOSPECIAL, OPERATIONS, NOOPERATIONS, AUDITOR, and NOAUDITOR) • LISTGRP (only group-name operand) • REMOVE (all operands)
JOIN	A user with JOIN group authority can define new users and groups to RACF and assign any level of group authority to new users (including the JOIN authority). To define new users, the user with JOIN authority must also have the CLAUTH user attribute for the USER class. When a user defines a new group, it becomes a subgroup of the group in which the user has JOIN authority. JOIN authority includes the privileges of USE, CREATE, and CONNECT authorities.	All of the above, plus: • ADDGROUP (all operands) • ADDUSER (all operands except OPERATIONS, SPECIAL, and AUDITOR) • ALTGROUP (SUPGROUP operand only; to change the superior group of a group, a user must have JOIN authority in one group and be owner of or be connected with the group-SPECIAL attribute to another group) • DELGROUP (all operands) • LISTGRP (only group-name operand)

For a list of the RACF commands that the group authorities allow users to issue, see Table 45 on page 330.

Suggestions for Assigning Group Authorities

As a security or group administrator, you can create different types of administrative structures, according to how you assign group authorities and group ownership. Two examples of possible structures are:

- **Total Delegation:** You can have one delegate (group owner) be responsible for the administration of a group, the users in the group, and the group resource profiles. The group owner, in this scheme, connects to the group with JOIN authority, defines the group resource profiles, and connects other users to the group with USE authority.
- **Partial Delegation:** You can share the responsibility for the administration of a group, the users in the group, and the group resource profiles. Under this scheme, the owner of the group connects one user to the group with JOIN authority and this user connects other users to the group, giving CREATE authority to one user and USE authority to all other users. In this way, the owner of the group can monitor the group, the user with JOIN authority can monitor the users in the group, and the user with CREATE authority can create and maintain the group's data set profiles.

Group Terminal Option

The group administrator (that is, the owner of a group) can specify a group terminal option for the group by using the ALTGROUP command with the NOTERMUACC operand. With this option, users of the group are authorized to log on to VM only from those RACF-protected terminals to which they have been specifically authorized access by the PERMIT command. That is, users of the group may not be authorized to log on to VM from terminals (either RACF-defined or otherwise) based on the universal access authority of the terminals.

Summary of Steps for Defining a RACF Group

This summary presents the steps required by RACF for defining a RACF group. Your installation may require additional steps, depending on your security policy.

Step 1 Prepare to create the group profile as follows:

- Decide which group is to be the superior group.
- Decide the group name.

  Note: This cannot be the same as a user ID.
- Decide who (a user or RACF group) is to be the owner of the new group. (If the group owner is a user, give him or her the information needed to manage the group.)
- If your installation is using RACF to protect terminals, and the users in this group are terminal users who are to be restricted to specific terminals, consider specifying the NOTERMUACC operand on the ADDGROUP command.

Step 2 Create the group profile using the ADDGROUP command.

For example, to create a group for Department A called DEPTA whose owner and superior group is to be a group called ALLDEPT, enter:

```
ADDGROUP DEPTA OWNER(ALLDEPT) SUPGROUP(ALLDEPT)
```

Step 3 Connect appropriate users to the new group.

- Most users should have USE group authority.
- A few users might need a group authority higher than USE group authority (such as CONNECT).

For example, to connect department members STEVEH, LIZS, and GENEK to the DEPTA group and also give LIZS and STEVEH authority to add new users to the group, enter:

```
CONNECT (STEVEH LIZS) GROUP(DEPTA) OWNER(DEPTA) AUTHORITY(CONNECT)
CONNECT GENEK GROUP(DEPTA) OWNER(DEPTA)
```

These commands assign ownership of each connection to group DEPTA rather than to the issuer of the CONNECT command (the default). Because GENEK's authority defaults to USE, GENEK can use any of the resources (for example, terminals) that belong to group DEPTA.

Step 4 If the group requires access to RACF-protected resources, give the group the required access using the PERMIT command. For example:

```
PERMIT TERM007 CLASS(TERMINAL) ID(DEPTA) ACCESS(READ)
```

Step 5 If the group requires access to OpenExtensions VM resources, alter the profile to include an OVM segment with a GID. For example:

```
ALTGROUP DEPTA OVM(GID(100))
```

Summary of Steps for Deleting Groups

This summary presents the steps required by RACF and related IBM program products to delete groups from RACF on VM. Your installation may require additional steps, depending on your security policy and the products you have installed.

Step 1 Remove all users from the group.

Note: You can use the REMOVE command to do this. Before removing a user from the user's default connect group, you must first connect the user to a new group (CONNECT command), then change the user's default connect group to the new group (ALTUSER command).

Step 2 To research the following steps, use the IRRUT100 utility to list the occurrences of the group ID in the RACF database.

Step 3 For each subgroup of the group to be deleted, change the subgroup's superior group to an existing group.

```
ALTGROUP  subgroup-name  SUPGROUP(new-superior-group-name)
```

Step 4 If the group is the owner of any profiles (the group's group ID was specified on the OWNER operand), change the owner of the profiles to a new group or user.

Note: To do this, use the appropriate RACF alter command, such as ALTUSER, ALTGRP, or RALTER.

Step 5 Remove the group from any access lists the group's group ID might be specified in.

Note: To do this, use the DELETE operand on the PERMIT command.

Step 6 If the group has a GID assigned to it in the OVM segment of its GROUP profile, check for any files in the BFS that are owned by the GID. These files will either need to be deleted or transferred to another GID using the **chgrp** or **chown** command. See *OpenExtensions for z/VM: Command Reference* for more information.

If the GID owns any set-GID executable files, there may be profiles in the VMPOSIX class that need to be deleted or renamed appropriately. These profiles have the format:

```
EXEC.Ggid
```

See “Protecting Set-UID and Set-GID Executable Files” on page 253 for details on these profiles.

Step 7 After all occurrences of the group ID are deleted from the RACF database, delete the group profile.

Note: This can be done using the DELGROUP command.

Chapter 5. Defining Profiles for General Resources

Table 10 lists the RACF commands you can use to work with general resource profiles.

Table 10. RACF Commands Used to Work with General Resource Profiles

Activity	**Command**
Defining	RDEFINE
Changing	RALTER
Allowing or denying access	PERMIT with CLASS(*class-name*)
Searching	SEARCH with CLASS(*class-name*)
Listing	RLIST
Deleting	RDELETE

Note: For the authority needed to issue any of these commands, see *RACF Command Language Reference*.

Summary of Steps for Defining General Resource Profiles

This summary presents the steps required by RACF to define general resource profiles. Please note that specific instructions are printed elsewhere in this book for most of the kinds of resources supported by IBM-supplied general resource classes.

1. Determine which resources are to be protected by the profile. This involves the following information:

   - The general resource class, such as TAPEVOL or TERMINAL.
   - The profile name:
     - If you specify a generic profile name, the profile can protect more than one resource.

       Using generic profiles instead of discrete profiles can greatly reduce the effort of maintaining the profiles. In general, you should create generic profiles to cover the majority of resources, using discrete profiles only for exceptions.

       Also, you should consider creating a profile to be used as a model, especially if you are specifying complex access lists. Models can be used when creating any kind of resource profile (discrete or generic), and modeling can be done across classes (to model, specify the FROM operand on the RDEFINE command; to model across classes, you should also specify the FCLASS operand). Before using modeling, see "Possible Changes to Copied Profiles When Modeling Occurs" on page 46.

       Note: To specify generic profile names, generic command processing or generic profile checking (the SETROPTS GENCMD or SETROPTS GENERIC option) must be in effect for the class. For example, for the TERMINAL class:

       ```
       SETROPTS GENERIC(TERMINAL)
       ```

     - If you specify a discrete profile name, the profile can protect only one resource.
     - For most IBM-supplied classes, the rules for specifying profile names are described in this book.

     Notes:

     a. For some kinds of resources, such as terminals, you should consider using resource group profiles instead of generic profiles. Creating resource group profiles can save a significant amount of work. See Chapter 8, "Creating Resource Group Profiles" on page 129 for more information.

     b. You can use RACFVARS profiles to specify which values will be taken by variables (indicated by an &) in profile names. See "Using RACF Variables in Profile Names (RACFVARS Class)" on page 99 for more information.

   - Decide which access is to be allowed to all users on the system who are not otherwise restricted. In RACF, this is called the universal access authority (UACC). This has the same meaning as the access authority on access lists (see step 3 on page 92). In most cases, the UACC should be NONE or READ.

- Decide which user or group is to be the owner of the new resource profile. By default, this is the user who creates the profile.
  - If the owner is a user, the owner will be able to list, modify, or delete the resource profile. Note that being the owner of a resource profile does not, by itself, allow a user to have access to the resource(s) protected by the profile. For more information, see step 11 in "Authorizing Access to Resources Protected by RACF Profiles" on page 307.
  - If the owner is a group, the authority of a user who has a group-level attribute in that group (such as group-SPECIAL or group-AUDITOR) extends to resources protected by this profile.
- Decide which user, if any, should be notified by a message when users make unsuccessful attempts to access resources protected by the profile (NOTIFY operand).
- Decide whether RACF should log access attempts to resources protected by the profile (AUDIT operand).

  Note: To see the results of the logging done by RACF, use the RACF SMF data unload utility or the RACF report writer. For more information, see *RACF Auditor's Guide*.
- If your installation is using some form of security classification, do one of the following:
  - If security labels are used on your system, decide which security label (if any) to assign to the profile.

    When security labels are being used on your system, be aware that security levels and categories are ignored.
  - If security levels are used on your system, decide which security level (if any) to assign to the profile.
  - If security categories are used on your system, decide which security categories (if any) to assign to the profile.
  - If your installation has written RACF installation exits to use the LEVEL operand, decide which value to specify for LEVEL.
- Depending on the class of the resource, the profile might have specific fields for which you should assign values. For example:
  - Profiles in the APPCLU class have SESSION segments.
  - Profiles in the TAPEVOL class have the SINGLEDS and TVTOC operands.
  - Profiles in the TERMINAL and GTERMINL classes have the WHEN and TIMEZONE operands (both optional). WHEN determines the times and days a terminal may be used.

    Note: This WHEN is not the same as the WHEN operand in a conditional access list.

  For specific information on these operands, see the appropriate section of this book, or see the description of the RDEFINE command in *RACF Command Language Reference*.

- To copy an existing profile, specify the name of the existing profile on the FROM operand. If the existing profile is in a different class, specify FCLASS also.

2. Create the general resource profile using the RDEFINE command.

   ```
   RDEFINE  class-name  profile-name  other-operands
   ```

   Note: To change a general resource profile, use the RALTER command.

3. If specific users or groups are to have specific access to the profile, use the PERMIT command to create one or both of the access lists:

   - Each entry in the standard access list states which access (such as NONE or READ) a specific user or group has:

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

   - Each entry in the conditional access list states which access (such as NONE or READ) a specific user or group has, and also states which condition a user must meet to get the specified access:

     ```
     PERMIT profile-name CLASS(class-name)
            ID(userid or group) ACCESS(access-authority)
            WHEN(condition)
     ```

   Notes:

   a. Access authorities you can specify with UACC or specifically assign to users vary from class to class, and are described in the sections of this book that describe the specific classes.

   b. This book does not describe all classes. For descriptions of classes, see the class descriptor table (CDT) in *RACF Macros and Interfaces* Also, for some classes (the FACILITY class, for example), the access required by some resource managers to specific profiles is described in the documentation of those resource managers.

4. If you have not already done so, activate the resource class:

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

5. For performance benefits, consider doing one of the following:

   - Allow all users on the system to have access to the resource at some level (such as READ or UPDATE) by creating a global access checking table (GAC) entry that has a name similar to the new resource profile.

     See Chapter 6, “Setting Up the Global Access Checking (GAC) Table” on page 103.

   - Reduce I/O to the RACF database by requesting that RACF keep all profiles in the class in storage:

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

     Note: This is required for some classes.

   - On VM, where large numbers of profiles can consume too much system storage, keep only generic profiles (not discrete profiles) in storage:

     ```
     SETROPTS  GENLIST(class-name)
     ```

Choosing Between Discrete and Generic Profiles in General Resource Classes

When you are creating profiles in the general resource classes you can create either a discrete or a generic profile.

Choose a *generic profile* to protect more than one resource with the same security requirements.

Notes:

1. You can use the characters *, **, or % to specify in which way (if at all) the resources protected by the profile have identical characters in their names.
2. If you use the character & in a profile name, there must be a corresponding RACFVARS profile. See “Using RACF Variables in Profile Names (RACFVARS Class)” on page 99.

Choose a *discrete profile* to protect one resource with unique security requirements. The name of a discrete profile has no generic characters.

Choosing Among Generic, Resource Group, and RACFVARS Profiles

Table 11 gives some considerations for choosing among generic profiles, resource group profiles, and RACFVARS profiles.

Table 11. Choosing Among Generic, Resource Group, and RACFVARS Profiles

How to Choose	Reference
Use generic profiles when the names of the resources have logically matching characters.	“Generic Profile Names” and *RACF Command Language Reference*.
Use resource group profiles if the names of the resources do not have logically matching characters and there is a resource grouping class (such as GTERMINL).	Chapter 8, “Creating Resource Group Profiles” on page 129.
Use RACFVARS profiles if the names of the resources do not have logically matching characters and there is no resource grouping class.	“Using RACF Variables in Profile Names (RACFVARS Class)” on page 99.

Generic Profile Names

When You Can Specify Generic Profile Names

You can create a profile with a generic name when either of the following is true for the class of the profile:

- The SETROPTS GENERIC option is in effect. Not only does this option allow the creation of generic profiles, it also causes RACF to use generic profiles during authorization checking.

- The SETROPTS GENCMD option is in effect. In this case, generic profiles can be created and modified, but RACF does not use them during authorization checking. This is intended for use when migrating from discrete profiles to generic profiles.

For generic profile naming of general resources, you can use an asterisk (*) within a profile name, representing one qualifier of a resource name. You can also use a double asterisk (**) to represent zero or more qualifiers within a general resource generic profile or at the end of such a profile.

Rules for Generic Profile Names

The following rules apply to profile names:

- Valid generic characters are *, %, and **:
  - Specify % in the profile name to match any single non-blank character (except a period) in the same position of the resource name.
  - Specify * or ** in the profile name to match more than one character in the same position of the resource name. For a complete description, with examples, of how to specify * and **, see *RACF Command Language Reference*.

Restricted Use of %* in General Resource Profile Names

The %* combination requires special attention.

New profiles with an ending %* are no longer allowed nor are profiles named %*. The RDEFINE command will return an error message.

Existing profiles with an ending %* are usable, but they should be deleted before creating any new profiles with a middle or beginning * or **. The RALTER and RDELETE commands will accept %* to enable you to make the changes.

Instead of using an ending %*, create new profiles ending with * for similar function (change `AB.C%*` to `AB.C*`).

If you have an existing profile whose entire name is %*, you should create a new profile whose new name is **.

Notes:

1. The above considerations also apply to generic members of grouping classes.
2. When creating the new profiles, consider using the FROM operand for continued use of the same access list.

- For any particular general resource class, the profile naming conventions are defined by how the resource name is specified on the call to RACF. When your application programmers are designing the resource names they will use in their invocations of the security product, they should be aware of the problems they will have in using *, %, or & in resource names. For more information, refer to *External Security Interface (RACROUTE) Macros for MVS and VM*.

  As you define general resource profiles, users must observe the naming conventions for that particular class. For some classes, the naming conventions are described in this book. However, other products (both IBM and non-IBM) can issue RACROUTE REQUEST=AUTH. You must check the documentation produced for those products for authoritative information on how those products call RACF. You should be able to gather the following information from the calling product's documentation:

- When the call to RACF is done. In other words, what user action causes the call to RACF?

  Some further questions to ask: Also, are there settings in the product that cause the call to occur? Are there installation exits that can prevent the call, or change the results of the call?

- What is the class name used on the call to RACF?
- What is the resource name used on the call to RACF? If you are using discrete profiles, this is the profile name. If you are using generic profiles, you need to know how many qualifiers (portions of the name that are separated by periods) there are, and what the qualifiers mean, so you can specify meaningful profile names.

  Note: If you do not follow the resource naming convention established by the caller of RACF, you could create profiles that are never used. For example, if you create a discrete profile with less than the correct number of qualifiers, the profile will never be used during RACF authorization checking.

- What do the access authorities (READ, UPDATE, CONTROL, ALTER) mean? Remember, these values are hierarchical (UPDATE is higher than READ, and so forth), and do not necessarily mean what the English word means. For example, for terminals, READ means “allowed to log on,” not “allowed to read information.”

Generic Profile Checking of General Resources

The rules for access-authorization checking of generic profiles for general resources are similar to those for the DATASET class.

- Generic profiles are not checked unless generic profile checking is in effect for the class. To do this, issue the following command:

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

- If the class is not active, RACF does not check for profiles. RACF returns the default return code of the class to the resource manager. For a complete description, see “Authorization Checking for Resources Protected by RACF Profiles” on page 306.
- If more than one profile covers a particular resource, RACF searches for profiles in the following order:

  1. Discrete profile
  2. Matching generic profiles (see Table 12 on page 96)

  and stops at the first matching profile.

Table 12. Sample General Resource Profile Names (Most Specific to Least Specific)

		Resources Being Accessed			
Profile Name	**Profile Type**	**MEDIUM**	**MEDIUM. PAPER**	**MEDIUM. PAPER. TEST**	**MEDIUM. ONLINE. FINAL**
MEDIUM. A	Discrete				
MEDIUM.ONLINE.FINAL	Discrete				X
MEDIUM.ONLINE.*	Generic				X
MEDIUM.PAPER	Discrete		X		
MEDIUM.PAPER.TEST	Discrete			X	
MEDIUM.PAPER.%	Generic				
MEDIUM.PAPER.*	Generic			X	
MEDIUM.PAPER.**	Generic		X	X	
MEDIUM.PAPER%	Generic				
MEDIUM.PAPER*	Generic		X	X	
MEDIUM.PAPE%	Generic		X		
MEDIUM.PAP*	Generic		X	X	
MEDIUM.PRINT.*	Generic				
MEDIUM.&X (where &X = PAPER in RACFVARS profile)	Generic		X		
MEDIUM.&Y (where &Y = ONLINE.FINAL in RACFVARS profile)	Generic				X
MEDIUM.%APER	Generic		X		
MEDIUM.*.FINAL	Generic				X
MEDIUM.*.FINAL*	Generic				X
MEDIUM.**.FINAL	Generic				X
MEDIUM.**.PAPER	Generic		X		
MEDIUM.*	Generic		X	X	X
MEDIUM.**	Generic	X	X	X	X
MEDIUM*.**	Generic	X	X	X	X
.	Generic		X	X	X
*.**	Generic	X	X	X	X
*	Generic	X	X	X	X
**	Generic	X	X	X	X

To determine which profiles have the potential to protect any particular resource, use the FILTER or MASK operands on the SEARCH command to generate a list of profiles that might match the resource. For example, you might specify the user's user ID on the FILTER operand to limit the list of profiles displayed, as follows:

```
SEARCH CLASS(VMMDISK) FILTER(**.userid.**)
```

In general, the list of profiles generated by the SEARCH command is the order in which RACF searches for a matching profile. To review the list, take the following steps:

1. Find all profiles that match the resource name.
2. If no profile names match, check for profile names that include & (RACF variables). You must list the RACFVARS profile to determine the value of a RACF variable:

```
RLIST  RACFVARS  variable-name
```

Also, the SEARCH command will not list grouping profiles (such as GTERMINL) that protect the resource. To do this, use the RESGROUP operand on the RLIST command.

```
RLIST  member-class resource-name  RESGROUP
```

See "Which Profiles Protect a Particular Resource?" on page 131.

If these methods do not find a profile, the resource is not protected.

3. If only one profile matches, it protects the resource.
4. Otherwise, find two profiles that both match the resource name. Then, compare them character by character. Where they first differ, if one has a discrete character, and the other has a generic character, the one with the discrete character wins. If both have a generic character where they differ, then:
   - If one has an &, and the other has a %, *, or **, the & wins.
   - If one has a % and the other has a * or **, the one with % wins.
   - If one has a * and the other has a **, the one with * wins.

Note: The following is generally true:

Given two generic profiles that match a resource, the one with a generic character farther from the beginning of the name is used.

Note: If two profile names match except for one character position, the following is the order in which RACF searches for them:

blank
.
$ (X'5B')
# (X'7B')
@ (X'7C')
A through Z
0 through 9
& (X'50')
%
*

For example, the following profile names all match in the first three character positions (A.B), and are shown in the order searched:

```
A.B
A.B.B
A.BA
A.BZ
A.B0
A.B9
A.B&X
A.B%
A.B*
```

When in doubt about the search order, create sample profiles and check the order of profile names shown by the SEARCH command.

Granting Access Authorities

You can grant (or deny) user or group access to a RACF-protected resource either explicitly, by assigning the specific user or group access authority with the appropriate command, or implicitly, with the universal access authority (UACC).

Each resource that you protect with RACF requires a UACC, which is the default access authority for the resource. All users in the system who are not specifically named in the access list of that resource profile can still access the resource with the authority specified by UACC (unless the UACC is NONE). These users include users not defined to RACF.

If you specifically assign a user or group an access authority to a resource, the specified authority overrides the UACC specified for the resource.

Valid authorities you can specify with UACC or specifically assign to users or groups vary from class to class, and are described in the sections of this book that describe the specific classes.

Note: Not all classes are described in this book (for example, the DSNR class is not described in this book). Also, in some classes, the access required by some resource managers to specific profiles is described in the documentation of the resource manager.

Table 13 shows additional meanings for several access authorities for general resources.

Table 13. Access Authorities for General Resources

Authority	Meaning
ALTER	For discrete profiles, the specified user or group has full control over the resource and the resource profile, and can authorize other users and/or groups to access the resource. For generic profiles, only the profile owner, users with the system-SPECIAL attribute, and group-SPECIAL users whose groups own the profile have control over the resource profile and can authorize other users and/or groups to access the resource. For both profiles, full resource access is allowed.
NONE	The specified user or group is not permitted to access the resource or to list the profile.
CONTROL READ UPDATE	These access authorities allow listing of selected portions of the profile and grant resource access in a variety of ways, depending on the class.

Conditional Access Lists for General Resource Profiles

You can require that a user or a job have entered the system from a particular device when accessing general resources. Specifically, you can require that a user be logged onto a particular terminal by specifying WHEN(TERMINAL(...)) on the PERMIT command.

The TERMINAL class must be active for this support to take effect.

Using RACF Variables in Profile Names (RACFVARS Class)

You can create profiles in the RACFVARS class whose profile names act like programming variables. The name of a RACFVARS profile can be specified as all or part of the names of general resource profiles that actually protect resources. The following sections describe ways in which you can make use of this facility.

- You can create a RACFVARS profile whose name is used as the entire profile name of profiles in the other class. On the ADDMEM operand used in creating the RACFVARS profile, you specify which resources are protected by the other profile.

  For example, a TAPEVOL profile name has only one qualifier, which is the tape volume being protected. You can create a RACFVARS profile named &TAPEV35, and specify several tape volumes on the ADDMEM operand. If a TAPEVOL profile has the same name as the RACFVARS profile (&TAPEV35), the TAPEVOL profile will protect those tape volumes.

- You can create a RACFVARS profile whose name is used as *part of* the profile name of profiles in the other class. The RACFVARS profile defines which values can be used for that part of the profile name.

- PERMIT commands issued for a RACFVARS profile affect the administration of that profile, not access to the resource protected with the RACFVARS name. For example, to allow users access to tape volumes protected by profile &TAPEV35 change the profile in the TAPEVOL class, not the profile in the RACFVARS class.

Using RACFVARS Profiles to Protect Many Resources

Like resource group classes (such as GTERMINL), you can use the RACFVARS class to create profiles that protect many resources that have *unlike* names.

To do this, take the following steps:

1. Create a profile in the RACFVARS class, specifying the resource names on the ADDMEM operand.

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

   where:

   profile-name must begin with the character &, can be up to 8 characters long (including the &) and cannot contain the characters *, %, or . (period). Profile names beginning with RAC are reserved for IBM use.

   UACC(READ) is specified to allow any user to use the RLIST command to determine how a particular variable is defined.

   resource-name can be up to 39 characters long and should not contain the characters *, %, or & because members containing those characters will not be effective.

   For example, if you wish to use &PAYJOB as a qualifier in profile names in the JESSPOOL class, you could create a profile named &PAYJOB:

   ```
   RDEFINE  RACFVARS  &PAYJOB  UACC(NONE)
            ADDMEM(TAXES CHECKS)
   ```

In any profile where you specify &PAYJOB, RACF uses TAXES and CHECKS for that qualifier.

2. Create another profile in another class with a profile name that includes the RACFVARS profile as part of its name. For convenience's sake, this profile is called the *protecting profile*, and its class is called the *protecting class*.

   ```
   RDEFINE  protecting-class  profile-name  UACC(NONE)
   ```

   Where part of profile-name has been defined in the RACFVARS profile defined in step 1 on page 99. For example:

   ```
   RDEFINE  JESSPOOL  NODEA.*.&PAYJOB.*.*.OUTPUT  UACC(NONE)
   ```

   Note: These examples assume that a SETROPTS GENERIC(*classname*) was previously issued to turn generics on for this class and that a SETROPTS REFRESH was then done.

   If the &PAYJOB profile contains members TAXES and CHECKS, then creating the JESSPOOL profile above is equivalent to:

   ```
   RDEFINE  JESSPOOL  NODEA.*.TAXES.*.*.OUTPUT  UACC(NONE)
   RDEFINE  JESSPOOL  NODEA.*.CHECKS.*.*.OUTPUT  UACC(NONE)
   ```

   In a profile name, a variable name is ended by the eighth character, the end of the profile name, or one of the following characters — whichever occurs first:

   . (period)
   another &
   %
   *

   For example, when using the RACFVARS class, you could enter:

   ```
   RDEFINE RACFVARS &ABCDEFG ADDMEM(A B)
   ```

   In this case, X.&ABCDEFGY.Z matches both X.AY.Z and X.BY.Z.

3. Permit users to the second profile, as needed:

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

   For example:

   ```
   PERMIT  NODEA.*.&PAYJOB.*.*.OUTPUT  CLASS(JESSPOOL)  UACC(NONE)
           ID(JOE TOM)  ACCESS(READ)
   ```

   Note: Giving access authority to the RACFVARS profile does not affect access to the resources. (You might, however, give a user ALTER access authority to allow the user to change the RACFVARS profile.)

4. When you are ready to start using the protection defined in the RACFVARS profiles, activate the RACFVARS class and activate SETROPTS RACLIST processing for the RACFVARS class:

   ```
   SETROPTS  CLASSACT(RACFVARS)  RACLIST(RACFVARS)
   ```

   Notes:

   a. Activating the RACFVARS class may be required to allow certain other functions, such as JES spool reload, to work correctly.

   b. If the RACFVARS class is already active, omit the CLASSACT operand.

   c. Any time you change the value of a RACF variable (by specifying the ADDMEM or DELMEM operand on the RALTER command), you must refresh SETROPTS RACLIST processing for first, the RACFVARS class and, second, any class that uses the RACF variable.

Chapter 6. Setting Up the Global Access Checking (GAC) Table

You can use global access checking to improve performance of RACF authorization checking for selected resources. Global access checking should be used for *public resources* that are accessed frequently. For example, an entry in the global access checking table can give all users on the system READ access to the MAINT 190 minidisk.

The global access checking table is maintained in storage and is checked early in the RACF authorization checking sequence. If an entry in the global access checking table allows the requested access to a resource, RACF performs no further authorization checking. This can avoid I/O to the RACF database to retrieve a resource profile, and can result in substantial performance improvements.

How Global Access Checking Works

When a user requests access to a resource for which a RACROUTE REQUEST=AUTH macro is issued, and global access checking is in effect for the class of the resource, and SETROPTS RACLIST processing is not in effect for the class, RACF searches the global access checking table for a matching entry. If there is a matching entry, RACF compares the access authority requested by the user (READ, UPDATE, CONTROL, or ALTER) to the access authority associated with the resource in the global access checking table.

If the requested access is less than or equal to the authority specified in the table entry for the resource, global access checking grants the requested access immediately, without checking the profile protecting the resource. Otherwise, normal RACF authorization checking is performed. Global access checking can only permit accesses, not deny them.

Attention

Because RACF performs global access checking before many of the other kinds of access authority checks, such as security label checking or access list checking, global access checking might allow access to a resource you are otherwise protecting. To avoid a security exposure to a sensitive resource, do not create an entry in the global access checking table for a resource protected by a profile containing a security level, security category, or security label. (If the security label in the profile is SYSLOW, a global access checking table entry with an access authority of READ can be created.)

Candidates for Global Access Checking

In planning the resources to be *public* through global access checking, consider the following:

- MAINT 190 system disk (CMS S-disk)
- MAINT 19E system disk extension (CMS Y-disk)
- ISPF minidisks
- OfficeVision minidisks
- Local system extension disks
- Disks for widely-used licensed program, such as APL
- Tools minidisks
- Virtual unit record devices for the RSCS service machine (VMRDR class)
- Frequently used RSCS nodes (VMNODE class)

For more information on protecting public minidisks, see *RACF System Programmer's Guide*.

Creating Global Access Checking Table Entries

To create an entry in the global access checking table, do the following:

1. Plan the entries for the global access checking table, using the following guidelines:

   a. Attempt to identify resource profiles that are accessed frequently and for which a performance benefit is desired.

b. Do *not* add entries to the global access checking table for profiles in classes that are RACLISTed. RACF will not search the global access checking table for profiles in RACLISTed classes.

c. If there are resource profiles with UACC other than NONE, consider adding similar entries to the global access checking table. Using this “matched pair” approach, each entry would have the same name as a profile, and the access specified in the entry would generally match the UACC of the profile. Do *not* add a global access checking table entry if any of the following are true:

   - The profile has a security level, security category, or security label (other than SYSLOW).

     Note: If the profile has a security label of SYSLOW, the global access checking 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 global access checking table entry.

   - The profile has an entry in a conditional access list that is *more restrictive* than the access level of the global access checking table entry.

   - The profile requests auditing of successful access attempts at or below the level specified in the corresponding global access checking table entry.

   For example, if you have a minidisk profile PROFS.399 with UACC(READ) and AUDIT(FAILURES(UPDATE)) specified, you might create a global access checking table entry for it as follows:

   ```
   RALTER GLOBAL VMMDISK ADDMEM(PROFS.399/READ)
   ```

   However, if there are users or groups in the standard access list of profile PROFS.399 with an access authority of NONE (which is lower than the UACC), do *not* create a global access checking table entry. A global access checking table entry would allow these users and groups to read the PROFS minidisk.

d. If you have resources protected by a generic profile with UACC other than NONE, and others protected by a more specific (generic or discrete) profile that has specific access requirements such as an access list, consider adding two entries: one for the larger set of resources (with access authority 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 MAINT.* with a UACC(READ), but you also have some specific profiles with more restrictive entries, such as, MAINT.191 with UACC(READ) and an access list with JOE/NONE, then create two entries:

   ```
   MAINT.191/NONE
   MAINT.*/READ
   ```

   The entry with /NONE will not fail any attempts but will stop requests for MAINT.191 from being granted by the MAINT.* entry.

See the examples later in this section for other possible entries.

2. Add the resource class to the global access checking table using the RDEFINE command with the GLOBAL operand and the class name:

```
RDEFINE  GLOBAL  class-name
```

3. To allow global access checking for a specific resource, add an entry to the global access checking table using the RALTER command as follows:

   ```
   RALTER GLOBAL class-name ADDMEM(resource-name/access-level)
   ```

   where:

   resource-name is the equivalent of a profile name in the class specified. If generic command processing is in effect for *class-name* (this is done with the SETROPTS GENCMD command), *resource-name* on the ADDMEM operand can include the generic characters *, **, or %. In general, the rules for specifying these characters are the same as the rules for specifying these characters in generic profile names except that generic characters are allowed in any qualifier (even if not allowed in certain qualifiers of the profile names).

   Once added, generic entries in the global access checking table are used in global access checking even if generic profile checking is turned off (SETROPTS NOGENERIC command).

   The resource name can also include the name qualifiers &RACUID (RACF user ID) or &RACGPID (*current connect group*—see Glossary). For example, the following entry allows users to have ALTER access to minidisks that begin with their own user IDs:

   The resource name can also include variables defined in the RACFVARS class.

   ```
   RALTER GLOBAL VMMDISK ADDMEM(&RACUID.*/ALTER)
   ```

   Note: The preceding entry does not *change* a user's access to his or her own minidisks, but speeds the process by which RACF grants the access. (It also prevents any auditing of such access attempts.)

   The word &RACGPID allows the user's current connect group to be used in the same way. For example, the following allows all users to have READ access to group data sets for their current connect group:

   ```
   RALTER GLOBAL DATASET ADDMEM('&RACGPID.**'/READ)
   ```

   Note: If the current connect group is found in the global access table and list-of-groups processing is in effect, list-of-groups checking is ignored.

   access-level can be NONE, READ, UPDATE, CONTROL, or ALTER.

   See *RACF Command Language Reference* for more information on specifying the ADDMEM option.

4. When you are finished updating the global access checking table, issue the SETROPTS command with the GLOBAL operand for each class affected:

   ```
   SETROPTS  GLOBAL(class-name)
   ```

   Attention

Save a listing of the global access checking table. This can assist you in recovering from the accidental deletion or alteration of the global access checking table or its entries. You can use the RLIST command to make this listing quickly.

It is recommended that you write an EXEC containing the commands used to create the global access checking table. The EXEC should include the RLIST command to provide an independent record of the actual table created. Also, if the global access checking table is accidentally deleted (using the RDELETE command), the EXEC can readily be used to regenerate the table.

5. It is strongly recommended for all classes that, for each entry in the global access checking table, you create a similar resource profile. Such a “matched pair” approach can help ensure the continuation of protection if global access checking becomes disabled. For example,

   ```
   RDEFINE  class-name  resource-name  UACC(access-level)
   ```

   At the end-of-volume (EOV) processing, RACROUTE REQUEST=AUTH is issued with OLDVOL specified for authority checking with the DATASET and TAPEVOL classes. This check bypasses the global access table checking and uses resource profile definitions for authority checking. By not having a “matched pair” approach, you may get different results.

Adding an Entry to the Global Access Checking Table

To add an entry to the global access checking table, issue the RALTER command with the ADDMEM operand, then refresh the in-storage global access checking table. For example:

```
RALTER  GLOBAL  class-name  ADDMEM(resource-name/level)

SETROPTS  GLOBAL(class-name)  REFRESH
```

Deleting an Entry from the Global Access Checking Table

To delete an entry from the global access checking table, issue the RALTER command with the DELMEM operand, then refresh the in-storage global access checking table. For example:

```
RALTER  GLOBAL  class-name  DELMEM(resource-name/level)

SETROPTS  GLOBAL(class-name)
```

Attention

Do not use the RDELETE command unless you intend to delete the entire global access checking table for that class.

Stopping Global Access Checking for a Specific Class

To stop global access checking for a specific class, issue the following command:

```
SETROPTS  NOGLOBAL(class-name)
SETROPTS  GLOBAL  REFRESH
```

Listing the Global Access Checking Table

To list the global access checking table, do one of the following:

- For a list showing the entries in a particular class, enter the following:

  ```
  RLIST  GLOBAL  class-name
  ```

  This shows the entries in the order in which they are searched by RACF.

- For a list that shows all entries in the table, see the DSMON report that lists the global access checking table (described in *RACF Auditor's Guide*).

Examples for VM Systems

Example 1: MAINT Minidisks

To allow all users to have READ access to all minidisks owned by MAINT except the MAINT 191 minidisk, do the following:

```
    SETROPTS  GLOBAL(VMMDISK)

    RDEFINE  GLOBAL  VMMDISK

(1) RALTER  GLOBAL  VMMDISK  ADDMEM(MAINT.*/READ)
(2) RALTER  GLOBAL  VMMDISK  ADDMEM(MAINT.191/NONE)

    SETROPTS  GLOBAL(VMMDISK)  REFRESH

(3) RDEFINE  VMMDISK     MAINT.*    UACC(READ)
(4) PERMIT   MAINT.*    CLASS(VMMDISK)  ID(SYSGROUP)  ACCESS(CONTROL)

(5) RDEFINE  VMMDISK     MAINT.191  UACC(NONE)
(6) PERMIT   MAINT.191  CLASS(VMMDISK)  ID(SYSGROUP)  ACCESS(CONTROL)
(7) SETROPTS GENLIST(VMMDISK)
```

The entry defined at (1) has the same name and access authority (READ) as the profile created at (3).

The entry defined at (2) has an access authority of NONE, which causes RACF to use a profile in the VMMDISK class. In this case, RACF uses the profile created at (5).

The access lists created at (4) and (6) allow SYSGROUP a higher authority (CONTROL) than the global access table entries (READ and NONE).

The command at (7) should be used if you are using generic profiles.

The following command would have no effect on USER1's attempts to LINK to the MAINT 190 disk, because the entry created at (1) protects this minidisk, and allows everyone READ access without checking a profile:

```
PERMIT   MAINT.190  CLASS(VMMDISK)  ID(USER1)  ACCESS(NONE)
```

To make such a PERMIT effective, you must also do the following:

```
RALTER  GLOBAL  VMMDISK  ADDMEM(MAINT.190/NONE)
```

Example 2: SFS Files and Directories

To allow all users to have ALTER access to their own SFS files and directories, do the following:

```
SETROPTS GLOBAL(FILE DIRECTRY) GENERIC(FILE DIRECTRY)
RDEFINE GLOBAL (FILE DIRECTRY)
RALTER GLOBAL DIRECTRY ADDMEM(*.&RACUID.**/ALTER)
RALTER GLOBAL FILE ADDMEM(*.&RACUID.**.*.*/ALTER)
```

To allow all users to have READ access to an SFS directory named POOL1:TOOLS.GENUSER and to the files within the directory, do the following:

```
RALTER GLOBAL DIRECTRY ADDMEM(POOL1.TOOLS.GENUSER/READ)
RALTER GLOBAL FILE ADDMEM(POOL1.TOOLS.GENUSER.*.*/READ)
```

Note: You must use the RACF format of SFS names for defining entries in the global access table. See For more information, see “RACF Format for SFS Directory and File Names” on page 260.

Example 3: The VMBATCH Class

If the VMBATCH class is active, you should consider specifying the following:

```
RALTER GLOBAL VMMDISK ADDMEM(&RACUID.*/ALTER)
```

This allows batch machines to access the minidisks of any user who submits batch requests. For more information, see “Protecting Alternate User IDs” on page 172.

Special Considerations for Global Access Checking

When using global access checking, consider the following:

- While RACF is searching the global access checking table for a matching entry, profiles in the class are ignored. If no global access checking 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.
- RACF searches the global access checking table for an entry that best matches the name of the resource, much as RACF searches for a matching profile. The output from the RLIST command shows the order used.
- 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.
- RACF bypasses global access checking if the PROFILE, CSA, or PRIVATE operand is specified on the request for RACF authorization checking (RACROUTE REQUEST=AUTH or RACHECK macro).
- Updated global access checking table entries become effective with the next IPL or after execution of the SETROPTS command with the GLOBAL(*class-name*) operand (with or without the REFRESH operand).
- The only use for an access of NONE in the global access table is to force RACF to look for a profile. This would typically be used when you have access list entries which have a lower access level than a data set's UACC, or when

you want to assure that auditing or security classification checking will take place for a specific data set.

Chapter 7. Security Classification of Users and Data

Security classification of users and data allows installations to impose additional access controls on sensitive resources. Each user and each resource can have a security classification in its profile. You can choose among the following:

- Security levels, security categories, or both
- You can use security labels, which are a combination of security levels and security categories, and are easier to maintain

A *security level (SECLEVEL)* is an installation-defined name that corresponds to a numerical security level (the higher the number, the higher the security level). A *security category (CATEGORY)* is an installation-defined name corresponding to a department or an area within an organization in which the users have similar security requirements. A *security label (SECLABEL)* is an installation-defined name corresponding to a security level and zero or more security categories.

This section will discuss security levels and security categories first. Security labels are discussed later (see “Understanding Security Labels” on page 117).

Effect on RACF Authorization Checking

Security classification processing takes place after global access checking (if active), but before RACF checks the standard access list. If global access checking does not allow access to the resource, RACF does security classification processing for any resource protected by a profile that has security category and/or security level data. (For information on global access checking, see Chapter 6, “Setting Up the Global Access Checking (GAC) Table” on page 103. For a complete list of the sequence of checks that RACF makes to grant or deny access to a resource, see “Authorization Checking for Resources Protected by RACF Profiles” on page 306.)

Attention

Because RACF performs global access checking before many of the other kinds of access authority checks, such as security label checking or access list checking, global access checking might allow access to a resource you are otherwise protecting. To avoid a security exposure to a sensitive resource, do not create an entry in the global access checking table for a resource protected by a profile containing a security level, security category, or security label (if the security label in the profile is SYSLOW, a global access checking table entry with an access authority of READ can be created). See “Authorization Checking for Resources Protected by RACF Profiles” on page 306.

Security Levels and Security Categories

Security classification processing consists of a two-step checking process that occurs when RACF is processing an authorization request. (Note that the SECDATA class must be active, the SECLABEL class must not be active, and the protecting resource profile must have security levels and/or security categories.)

1. RACF compares the security level of the user with the security level of the resource. If the resource has a higher security level than the user, RACF denies the request. For a terminal session, the security level that RACF uses for the user is the lower of the user's SECLEVEL and the terminal's SECLEVEL. Thus if the terminal has a SECLEVEL of 50 and the user has a

SECLEVEL of 100, the user cannot access, through that terminal, any data that has a SECLEVEL of over 50.

2. RACF compares the list of security categories in the user's profile with the list of security categories in the resource's profile. If RACF finds any security category in the resource profile that is not in the user's profile, RACF denies the request. If RACF does not deny the request, RACF continues with authorization processing. If there are no categories in the resource profile, RACF continues with authorization processing.

Security Labels

Security label authorization checking is dependent on the concept of controlling user access to resources on the basis of three factors:

1. The sensitivity of the data that the resource contains
2. The user's authorization to access information at that level of sensitivity
3. The purpose for which the user is attempting to access the resource.

The security administrator indicates the sensitivity of the data in the resource as well as the authorization of the user by assigning appropriate security labels in the resource or user profile.

Security label authorization checking involves comparing the user's security label with the security label of the resource. A user who lacks sufficient authorization is prevented from accessing information in the resource.

Three types of authorization checks are used to determine security label authorization:

- **Read Only (R/O):** A user is attempting to read information from a resource
- **Write Only (W/O):** A user is attempting to write information to a resource (with no reading)
- **Read and Write (R/W):** A user is attempting to access a resource for the purpose of both reading and writing.

For more information, see “Security Label Authorization Checking” on page 315.

Understanding Security Levels and Security Categories

When RACF is first installed, security classification of users and data is inactive. To use security levels and categories, activate the SECDATA class (but not the SECLABEL class).

You can choose to use one or both parts of security classification processing. To use security level checking, you must define a profile in the SECDATA general resource class with the name SECLEVEL. To use security category checking, you must define a profile in the SECDATA general resource class with the name CATEGORY. The installation names for security categories and security levels are then defined as members of these profiles (in a manner similar to the global access table entries). You maintain the member entries by using the ADDMEM operand on the RDEFINE command and the ADDMEM and DELMEM operands on the RALTER command.

In the CATEGORY profile, the member entries are the names of the security categories. In the SECLEVEL profile, each member entry consists of a security level name followed by its associated security level number.

Note: You cannot define a SECLEVEL for a SECLEVEL profile in the SECDATA class. As a result, RACF does not perform security level checking when determining a user's authority to access a SECLEVEL profile. Also, if you issue the RLIST SECDATA SECLEVEL command to display a SECLEVEL profile, RACF will not display values in the SECLEVEL or CATEGORY fields of the profile.

Defining and Maintaining Security Levels and Security Categories

To use security levels and categories, take the following steps:

1. Define the SECLEVEL profile to the SECDATA class using the RDEFINE command.

   ```
   RDEFINE  SECDATA  SECLEVEL  UACC(NONE)
   ```

2. Define security levels as members of the SECLEVEL profile in the SECDATA class.

   ```
   RALTER   SECDATA  SECLEVEL  ADDMEM(seclevel-name/seclevel-number ...)
   ```

3. Define the CATEGORY profile to the SECDATA class using the RDEFINE command.

   ```
   RDEFINE  SECDATA  CATEGORY  UACC(NONE)
   ```

4. Define categories as members of the CATEGORY profile in the SECDATA class.

   ```
   RALTER   SECDATA  CATEGORY  ADDMEM(category-1 category-2 ...)
   ```

5. Assign security levels, security categories, or both, to users:

   ```
   ALTUSER userid SECLEVEL(security-level-name)

   ALTUSER userid ADDCATEGORY(category-name1 category-name2 ...)
   ```

6. Assign security levels, security categories, or both, to resources, for example:

   ```
   RALTER class-name profile-name SECLEVEL(security-level-name)

   RALTER class-name profile-name
         ADDCATEGORY(category-name1 category-name2 ...)
   ```

7. When you are ready to start using security levels and security categories, activate the SECDATA class:

   ```
   SETROPTS CLASSACT(SECDATA)
   ```

Example Showing an Error and Its Correction

The following example illustrates an error in setting up security levels and how it can be corrected:

1. Define a profile named CATEGORY with member entries:

   ```
   RDEFINE SECDATA CATEGORY UACC(READ) ADDMEM(ACCOUNTING)
   ```

   This command creates a profile named CATEGORY in the SECDATA class with the entry ACCOUNTING in its member list.

The UACC specification means that anybody can list this profile, to determine what security category names the installation has defined.

2. Define a profile named SECLEVEL:

   ```
   RDEFINE SECDATA SECLEVEL UACC(READ)
   ```

   This command creates a profile named SECLEVEL in the SECDATA class.

3. Define members to SECLEVEL:

   ```
   RALTER SECDATA SECLEVEL
          ADDMEM(IMPORTANT/10,ROUTINE/75,CONFIDENTIAL/150)
   ```

   This command defines security level names and the associated security level numbers as members of the SECLEVEL profile. The members created are:

Security Level Name	Number
IMPORTANT	10
ROUTINE	75
CONFIDENTIAL	150

   However, you discover that you made an error when defining the members of the SECLEVEL profile. You really wanted IMPORTANT to have a higher security level value than ROUTINE. To change this value, see the following example.

4. Change a level number:

   ```
   RALTER SECDATA SECLEVEL ADDMEM(IMPORTANT/100)
   ```

   This command changes the security level number associated with IMPORTANT. The new member list is:

Security Level Name	Number
ROUTINE	75
IMPORTANT	100
CONFIDENTIAL	150

Attention

Any change to existing SECLEVEL and/or CATEGORY members may cause unexpected results, because the change is not reflected in existing user and resource profiles. Whenever you make such a change, RACF issues a warning message to remind you. Installations can use the SEARCH command to find profiles that require changes. However, because RACF keeps track of security levels by number, replacing an existing security level name does not affect the protection that the security level number provides. If you had defined the security levels shown in the preceding example and then replaced CONFIDENTIAL/150 with SECRET/150, a listing of a user or resource profile that included the security level 150 would show the new name. Because the security level number is the same, there is no need to change any resource or user profiles.

Note: The need to change many profiles when a security level or security category is changed can be avoided by using security labels instead.

CATEGORY and SECLEVEL Information in Profiles

The RACF commands for users, MVS data sets, and general resources allow you to define and maintain security classification information. Some examples of commands with security category and security level information follow. (These commands are fully documented in *RACF Command Language Reference*.) The examples assume that the SECLEVEL and CATEGORY tables shown earlier have been defined.

VM Examples

1. Protect a VM minidisk with security category and security level information.

   ```
   RDEFINE VMMDISK SMITH.191 SECLEVEL(ROUTINE)
          ADDCATEGORY(ACCOUNTING)
   ```

   This command creates a general resource profile with a security level of ROUTINE and a security category of ACCOUNTING.

2. Modify the security level information and add a category in a general resource profile for a VM minidisk.

   ```
   RALTER VMMDISK SMITH.191 SECLEVEL(CONFIDENTIAL)
         ADDCATEGORY(PERSONNEL)
   ```

   This command modifies the profile to have a security level of CONFIDENTIAL instead of ROUTINE and adds a security category of PERSONNEL.

The security classification information in a user's profile is an access allowance, while in a resource profile it is an access restriction. In this way, the security level test “passes” a user whose SECLEVEL is greater than or equal to that of the resource. A similar situation exists with security categories. The security category test “passes” a user if the user's profile contains every security category that is in the resource's profile. However, passing the security level and category tests does not allow the user to access the resource. The user must also pass any other existing test.

Security classification information in user and resource profiles can be updated at any time. However, changes made to a user's security classification while the user is logged on will not take effect during that session.

Note: Only users with the SPECIAL attribute can give another user, data set, or general resource a security level higher than they have, or a security category that they do not have themselves.

Converting from LEVEL to SECLEVEL

Many installations use the LEVEL field for their own implementation of security-level checking. To convert these profiles to use SECLEVEL instead, installations can use the SEARCH command to search for profiles that have a specified value in the LEVEL field.

Attention

Before converting from the use of LEVEL to SECLEVEL, all user profiles must have the appropriate SECLEVELs (if the SECDATA class is activated).

Deleting UNKNOWN Categories

If you delete a member from the CATEGORY profile, and that category is still specified in resource profiles, the resource profile listing (for example, done with the RLIST command) will show an UNKNOWN category. To delete this category, enter a command like the following (with no category specified on the DELCATEGORY operand):

```
RALTER  class-name  profile-name  DELCATEGORY
```

To search for such profiles, enter the search command as follows:

```
SEARCH  CLASS(class-name)  CATEGORY
```

Understanding Security Labels

You can use *security labels* to associate a specific security level with a set of (zero or more) security categories. Security labels, when associated with resources, users, and jobs, provide the following advantages over security levels and security categories:

- Security labels can be assigned to data that is not necessarily protected by a resource profile. For example, spool files are assigned the security label of their creators. In many cases, data that has been assigned a security label retains that security label from the time the data is created until the data is deleted. For example, when a spool file is created by a user or job that is running under a security label, the spool file is assigned the security label of the user or job. The spool file retains that security label until the spool file itself is deleted (which can be long after the user logs off or the job ends).
- Users can log on with different security labels at different times but with the same user ID; without security labels, a user always has the same (default) security level and categories.
- Output printed for a user or job by the Print Services Facility can have a PSF identification label related to the security label of the user or job printed on every page.
- Output printed for a user by the VM system or by RSCS Secure Printing can have an identification label related to the security label of the user printed for each output file.
- It is easier to maintain the security classification of users and data (changing the definition of a security label affects all users and resources that have that security label; you need not make the same change for many different profiles as you would for security levels and categories).

To know what system configuration you need to use security labels, refer to Page 11. The following are some considerations related to security labels:

- If you assign a security label to a resource profile and activate the SECLABEL class, you must also assign a default security label to users who access the resource(s) protected by that profile. Further, the security label you assign to the users must allow the user to access the resource. For information on assigning security labels to users, see “Assigning Security Labels to Users” on page 120.

- If your installation uses the SETROPTS MLACTIVE option, all data protected by classes that require security labels must have security labels. For more information, see “Enforcing Multilevel Security” on page 235.
- If your installation uses the SETROPTS MLS(FAILURES) option, there are tighter restrictions on attempted accesses to resources, depending on the kind of access attempt and the security labels of the resource and user. For more information, see “Security Label Authorization Checking” on page 315.
- If your installation uses the SETROPTS MLS(FAILURES) option, the first data set written to a tape volume defines the security label of any data set that is later written to the tape. For more information, see “Preventing Users from Copying Data from One Security Label to a Lower Security Label” on page 234.
- If your system includes products that do not support security labels when they invoke RACF, you should consider using the SETROPTS COMPATMODE option. See “Compatibility Mode for Security Labels” on page 234.

Creating a Security Label

To create a security label, you must create a profile in the SECLABEL class. The name of the profile is the security label.

Notes:

1. Any time you make a change to a SECLABEL profile, you must also refresh SETROPTS RACLIST processing for the SECLABEL class for the change to take effect. For example:

   ```
   SETROPTS  RACLIST(SECLABEL)  REFRESH
   ```

2. You need not activate the SECDATA class.

To create a SECLABEL profile, do the following:

1. Define the SECLEVEL profile to the SECDATA class using the RDEFINE command.

   ```
   RDEFINE  SECDATA  SECLEVEL  UACC(NONE)
   ```

2. Define security levels as members of the SECLEVEL profile in the SECDATA class.

   ```
   RALTER   SECDATA  SECLEVEL  ADDMEM(seclevel-name/seclevel-number ...)
   ```

3. Define the CATEGORY profile to the SECDATA class using the RDEFINE command.

   ```
   RDEFINE  SECDATA  CATEGORY  UACC(NONE)
   ```

4. Define categories as members of the CATEGORY profile in the SECDATA class.

   ```
   RALTER   SECDATA  CATEGORY  ADDMEM(category-1 category-2 ...)
   ```

5. For each security label, define a profile in the SECLABEL class. The profile names are the security labels available on your system, and must be no longer than eight characters. For each SECLABEL profile, specify a security level and (optionally) a set of categories. For example:

   ```
   RDEFINE SECLABEL security-label SECLEVEL(seclevel-name)
           ADDCATEGORY(category-1 category-2 ...)
   ```

6. Users cannot use a particular security label (for logging on, for submitting a job, or for specifying in a RACF profile) unless they have at least READ access authority to the SECLABEL profile of that name. For example, if the SECLABEL profile named EAGLE has UACC(NONE) specified, and you wanted user AHLEE and group GROUP1 to be able to log on with a security label of EAGLE, issue the following command:

   ```
   PERMIT  EAGLE  CLASS(SECLABEL)  ACCESS(READ)  ID(AHLEE GROUP1)
   ```

7. When you are ready to start using security labels, activate the SECLABEL class and activate SETROPTS RACLIST processing for the class. SETROPTS RACLIST processing is required for the SECLABEL class. You can do these two actions in one command:

   ```
   SETROPTS  CLASSACT(SECLABEL)  RACLIST(SECLABEL)
   ```

For more information on the commands used in this section, see *RACF Command Language Reference*. For VM considerations when the security label class is active, see “Activating the Security Label Class on VM” on page 189.

Relationship of SECLABEL to SECLEVEL and CATEGORY in Resource Profiles

If the SECLABEL class is active, any existing SECLEVEL and CATEGORY information in resource profiles will be ignored. Even though SECLEVEL and CATEGORY information will be ignored, you can maintain it (for example, using RACF commands). If the SECLABEL class is not active, then RACF will continue to use the SECLEVEL and CATEGORY function as it is currently implemented. Thus by activating or deactivating the SECLABEL class using the SETROPTS command, an installation can choose to use or ignore security labels.

Security Label Naming Restrictions

Security Labels SYSHIGH, SYSLOW, and SYSNONE

SYSHIGH, SYSLOW, and SYSNONE are security labels that you can specify for resource and user profiles, but that you need not, and indeed cannot, create directly. (At RACF initialization, RACF creates SECLABEL profiles with these names if they do not already exist.)

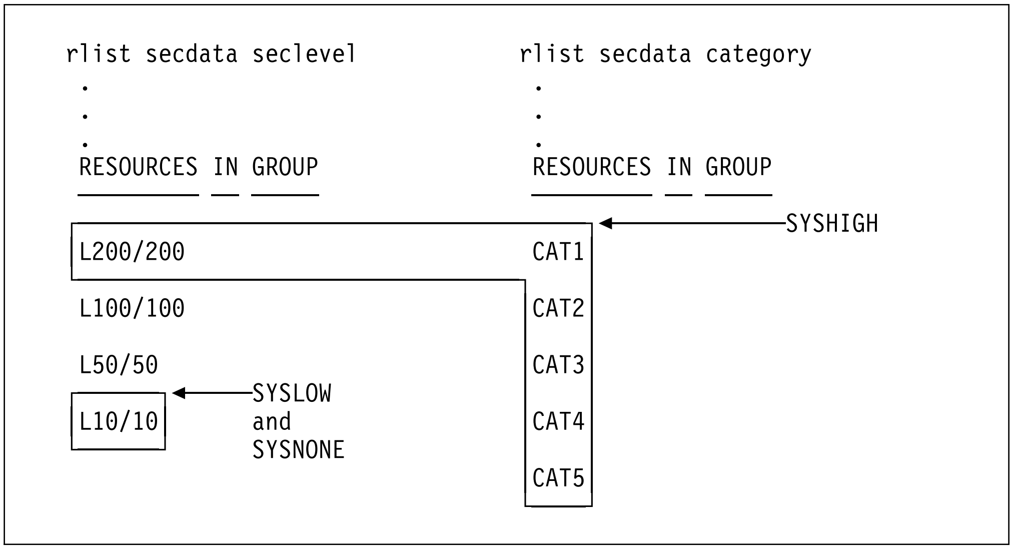
- SYSHIGH combines the highest security level specified in the SECLEVEL profile with all the categories defined in the CATEGORY profile.
- SYSLOW is the lowest security level specified in the SECLEVEL profile and *no* categories.
- SYSNONE is the same as SYSLOW, but is intended for use on resources that must be written to at different security labels when the SETROPTS MLS option is in effect (such as system catalogs).

These profiles do not actually contain the security levels and security categories that define SYSHIGH and SYSLOW. To find out what security levels and categories are used for security labels SYSHIGH and SYSLOW, do the following two commands:

```
RLIST  SECDATA  SECLEVEL

RLIST  SECDATA  CATEGORY
```

Figure 7 on page 120 illustrates the SYSHIGH and SYSLOW security labels. The left side shows part of the RLIST output for the SECLEVEL profile; the right side shows part of the RLIST output for the CATEGORY profile. With the security levels and categories currently defined on the system, SYSHIGH includes security levels L200 and categories CAT1 through CAT5. SYSLOW includes only security level L10 (no categories).



```
rlist secdata seclevel            rlist secdata category
 .                                 .
 .                                 .
 .                                 .
 RESOURCES IN GROUP                RESOURCES IN GROUP
 --------- -- -----                --------- -- -----

 L200/200                          CAT1

 L100/100                          CAT2

 L50/50                            CAT3

 L10/10                            CAT4

                                   CAT5
```

Figure 7. Sample SYSHIGH and SYSLOW Security Labels

The actual combination of security level and categories used to define the security labels SYSHIGH and SYSLOW is determined whenever the SETROPTS RACLIST command or SETROPTS RACLIST REFRESH command is issued for the SECLABEL class.

Security Label NONE

You should not define a security label on VM with the name NONE for following reasons:

- NONE is the default security label for VM system printers.
- VM does not permit jobs to print on a printer with a security label of NONE.
- When a spool file has no security label, the response to a QUERY READER, QUERY PRINTER, QUERY PUNCH or QUERY TRFILES command will contain the character string NONE in the SECLABEL field.

Assigning Security Labels to Users

On VM, just one security label (the default) can be stored in the user profile.

As security administrator, you should specify a default security label for each user that might need access to resources protected by security labels. To specify a user's default security label, enter the ADDUSER or ALTUSER command with the SECLABEL operand specified. You must also give each such user authority to use the security label. To give a user authority to use a security label, use the PERMIT command. For example:

```
ALTUSER  userid  SECLABEL(security-label)

PERMIT   security-label  CLASS(SECLABEL)
         ID(userid)  ACCESS(READ)
```

Notes:

1. On MVS, if you have a great many user profiles to which to add security labels, you can use the SEARCH command to generate a TSO CLIST that you can edit (for tailoring), then run. For example:

   ```
   SEARCH CLASS(USER) CLIST('ALTUSER ' 'SECLABEL(most-common-security-label)')
   ```

2. On VM, you can use the RACF ISPF panels to generate an EXEC (instead of a CLIST).

How Users Specify Current Security Labels

On VM, when a user logs on and does not specify a security label on the CP LOGON command, the default security label stored in the user profile becomes the user's current security label. The user can override the default security label by specifying the SECLABEL operand on the LOGON command as follows:

```
LOGON userid SECLABEL security-label
```

When you are migrating from security levels and security categories to security labels, you should consider setting the SECLABEL field using the ADDUSER and ALTUSER commands as follows:

```
ADDUSER  userid  SECLABEL(security-label)

ALTUSER  userid  SECLABEL(security-label)
```

Listing Security Labels

To display the security label stored in a resource profile, specify the ALL keyword on the LISTDSD and RLIST commands.

To display a user's *default* security label (the security label stored in the profile using the SECLABEL operand on the ADDUSER or ALTUSER command), issue the LISTUSER command with the user ID specified. For example:

```
LISTUSER  JONES
```

Displaying the Current Security Label for a User ID

You can use the RACSEC EXEC to display the current security label for your user ID or another user ID.

The syntax of the EXEC is as follows:

```
RACSEC {*|user ID}
```

*****	Use the asterisk (*) to specify that you want the current security label for your user ID.
user ID	Specify a user ID to obtain the current security label information for that user ID.

If you do not have a SECLABEL defined to your user ID, the following message is displayed:

```
RACSEC002I  Userid userid is not currently logged on, or does
            not have a security label.
```

If a SECLABEL is defined to your user ID, the following message is displayed:

```
RACSEC004I  The security label for user userid is seclabel.
```

To query the security label of a user other than yourself, you must meet one of the following requirements:

- If RACF protection for DIAGNOSE X'A0' subcode X'30' is in effect, then you must have READ access to the DIAG0A0.QUERYSEC profile in the VMCMD class.
- If RACF protection is not in effect for DIAGNOSE X'A0' subcode X'30', then you must have privilege class A or B.

For more information, see “Protecting the DIAGNOSE X'A0' Subcodes” on page 187.

Finding Out Which Security Labels a User Can Use

To find out which security labels a user can specify, issue the following command:

```
SEARCH  CLASS(SECLABEL)  USER(userid)
```

Note: Since SECLABELs must be RACLISTed, the class must be active before this SEARCH command can work.

Searching by Security Labels

To search for all profiles that have a particular security label, enter the following command:

```
SEARCH  CLASS(class-name)  SECLABEL(security-label)
```

For example:

```
SEARCH  CLASS(TERMINAL)  SECLABEL(EAGLE)
```

This command displays all the terminal profiles that have security label EAGLE specified.

```
SEARCH  CLASS(USER)      SECLABEL(EAGLE)
```

This command displays all the user profiles in which security label EAGLE is the default security label.

Notes:

1. You can search only one class at a time.
2. RACF lists only the names of profiles to which the user has at least READ access authority.

Restricting Security Label Changes

You can restrict users who do not have the SPECIAL attribute from specifying security labels in resource profiles, or changing the definitions of security labels, by specifying the SECLABELCONTROL operand on the SETROPTS command. For more information, see “Restricting Changes to Security Labels” on page 234. When the SECLABELCONTROL option is off, any user can specify the SECLABEL operand if the user has at least READ access authority to the associated SECLABEL profile.

Requiring Security Labels

You can require that all work entering the system, including users logging on and batch jobs, have a security label assigned. For more information, see “Enforcing Multilevel Security” on page 235. For what products must be installed, see Page 11.

SECLABEL Tranquility Considerations

For an evaluated B1 system, when the security label of a resource is changed, none of the affected resources should be in use. The same thing applies when implicitly altering the definition of the SECLABEL profile by changing the SECLEVEL or CATEGORY profiles in the SECDATA class, or by changing which security level or security categories apply to profiles in the SECLABEL class. For example, the following commands can implicitly change the definition of a security label:

```
RALTER  SECDATA  SECLEVEL  ADDMEM(...)
RALTER  SECLABEL  EAGLE  SECLEVEL(...)
RALTER  SECLABEL  EAGLE  ADDCATEGORY(...)
RALTER  SECLABEL  EAGLE  DELCATEGORY(...)
```

The inability of users to use any resources while either the users or resources security label is being changed is called *tranquility.*

For a list of what products must be installed, see Page 11.

Preventing any Changes to Security Labels

You can prevent any user from changing the security label of a profile or from changing a SECLABEL profile. For more information, see “Preventing Changes to Security Labels” on page 236.

Enforcing SECLABEL Tranquility

You can prevent users other than SPECIAL users, console operators, and started tasks from logging on, starting new jobs, or accessing resources. For more information, see “Temporarily Preventing Significant RACF Activity” on page 233.

Any change of the security label in a data set profile will be audited. In addition, if SETROPTS MLACTIVE is on, a type 83 SMF record is produced. This record contains the list of the cataloged data sets affected by the change, addition or deletion of a DATASET profile.

- If the SETROPTS CATDSNS option is in effect, this list is a correct list of all data sets affected by the security label change.

- If the SETROPTS CATDSNS option is not on, the list of the affected data sets may not be complete. Thus, the SETROPTS CATDSNS option allows you to list and audit the data sets affected by the change in a particular profile.

Planning Considerations for Security Labels

Even though a single user can use more than one security label, it may be easier to assign each person a separate user ID for each security label used.

Also, some consideration should be taken before using resource profiles that protect resources that need to be accessed when the user is logged on at different security labels (for example, a user's 191 minidisk).

Some products and applications depend on having a minidisk accessed in R/W mode to function properly. You should consider creating a minidisk and corresponding VMMDISK profile for each security label a user will be logging on with. By doing this, a user will always have at least one minidisk accessed in R/W mode when logged on.

For example, if SMITH needs to use security labels EAGLE and THRUSH, create VMMDISK profiles like:

```
SMITH.191   UACC(NONE)  SECLABEL(EAGLE)
SMITH.291   UACC(NONE)  SECLABEL(THRUSH)
```

Resources can be grouped into several categories and following certain procedures will minimize the impact of using them.

- **Read Only**

  In general, these resources pose no problem if they are protected by a profile with the lowest security label that a user will use. When protected in this manner, they can be accessed any time the user is logged on. For example, system resources that are read by all users should be protected with the SYSLOW security label.

- **Read Mostly**

  This type of resource should also be protected by a profile at the lowest security label that the user will use. This allows the user to access them for read any time the user is logged on. If the resource needs to be updated (for example, a file on a user's 291 minidisk) the user must log on at his or her lowest security label in order to update the file.

- **Read/Write**

  This type of resource should be protected by a profile at the security label that the user will most frequently use. When the SETROPTS MLS(FAILURES) option is in effect, access to these resources is prevented if the user is logged on with a security label different from the security label of the resource.

When RACF checks a user's authority to use a terminal, RACF uses “reverse MAC” (mandatory access checking). That is, the security label of the profile protecting the terminal must be equal to or greater than the security label of the user.

Security Labels: An Example

Figure 8 on page 126 shows how to set up security labels to meet the following needs:

- Projects on the same system cannot access each other's data
- Projects on the same system have access to common data (such as tools or data that they need read access to, but not write access)

In this example, there are two projects, A and B. Each project has one category (categories A and B). (While each project can have more than one category, it is simpler to show only one for each project.)

The system has four security levels:

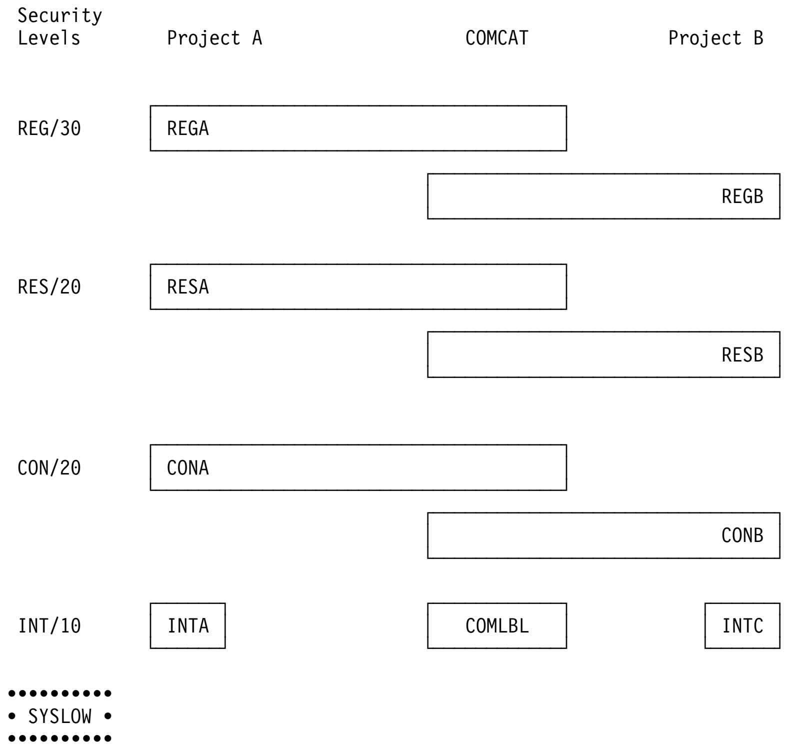
REG (registered, currently the highest security level on the system)
RES (restricted)
CON (confidential)
INT (internal)

To give each of the two projects access to common data, do one of the following:

- If the SETROPTS MLACTIVE option is in effect, create a third security category (called COMCAT). When you create security labels, include the common category in security labels to allow users logged on at those security labels to read common data.

  Create a security label at the lowest defined security level (INT) that includes a third security category (COMCAT).
- If the SETROPTS MLACTIVE option is *not* in effect, assign no security label to profiles protecting the common data.

Data at the lowest security level (INT) within a particular project is shown primarily to show the effect of creating security labels that include no category for common data.



Notes:

1. SYSHIGH includes security level REG and categories A, B, and COMCAT.
2. SYSLOW includes security level INT and no categories.

Figure 8 (Part 1 of 4). Security Labels: An Example

Note: The following commands assume that the security levels and categories have already been defined.

Commands to Define Security Labels for Project A:

```
RDEFINE SECLABEL REGA SECLEVEL(REG) ADDCAT(A COMCAT) UACC(NONE)

RDEFINE SECLABEL RESA SECLEVEL(RES) ADDCAT(A COMCAT) UACC(NONE)

RDEFINE SECLABEL CONA SECLEVEL(CON) ADDCAT(A COMCAT) UACC(NONE)

RDEFINE SECLABEL INTA SECLEVEL(INT) ADDCAT(A) UACC(NONE)
```

Commands to Define Security Labels for Project B:

```
RDEFINE SECLABEL REGB SECLEVEL(REG) ADDCAT(B COMCAT) UACC(NONE)

RDEFINE SECLABEL RESB SECLEVEL(RES) ADDCAT(B COMCAT) UACC(NONE)

RDEFINE SECLABEL CONB SECLEVEL(CON) ADDCAT(B COMCAT) UACC(NONE)

RDEFINE SECLABEL INTB SECLEVEL(INT) ADDCAT(B) UACC(NONE)
```

Command to Define the Common Security Label:

```
RDEFINE SECLABEL COMLBL SECLEVEL(INT) ADDCAT(COMCAT) UACC(NONE)
```

Commands to Assign Security Labels to Users:

Security administrator (a SPECIAL or group-SPECIAL user):

```
ALTUSER ADMIN SECLABEL(SYSHIGH)
```

A team leader has access to all security labels, but usually logs on to the highest security label in his or her project (REGA):

```
ALTUSER  LEADERA  SECLABEL(REGA)

PERMIT  (REGA RESA CONA INTA)  CLASS(SECLABEL)  ID(LEADERA)  ACCESS(READ)
```

A newly hired worker has access only to the lowest security label in his or her project:

```
ALTUSER  WORKERA  SECLABEL(INTA)

PERMIT  INTA CLASS(SECLABEL)  ID(WORKERA)  ACCESS(READ)
```

Note: Similar commands are issued for project B users.

Commands to Assign Security Labels to Data for Project A:

```
ALTDSD  'GROUPA.REGA.**'  SECLABEL(REGA)
ALTDSD  'GROUPA.RESA.**'  SECLABEL(RESA)
ALTDSD  'GROUPA.CONA.**'  SECLABEL(CONA)
ALTDSD  'GROUPA.INTA.**'  SECLABEL(INTA)

RALTER  VMMDISK  GROUPA.294  SECLABEL(REGA)
RALTER  VMMDISK  GROUPA.293  SECLABEL(RESA)
RALTER  VMMDISK  GROUPA.292  SECLABEL(CONA)
RALTER  VMMDISK  GROUPA.291  SECLABEL(INTA)
```

Note: Similar commands are issued for project B data.

Commands to Assign Security Labels to Common Data:

```
ALTDSD  'COMMON.DATA' SECLABEL(COMLBL)

RALTER  VMMDISK  COMMON.191  SECLABEL(COMLBL)
```

Figure 8 (Part 2 of 4). Security Labels: An Example

What Users Can Do Based on Security Label Authorization Checking (SETROPTS NOMLS in Effect):

Note: Other authorization requirements, such as access lists and UACC, can prevent users from accessing data. This figure is limited to the controls enforced by security label authorization checking (with the SECLABEL class active and the SETROPTS NOMLS option in effect).

Users who log on with project A security labels can only view data with security labels REGA, RESA, CONA, INTA, and COMLBL.

Users who log on with project B security labels can only view data with security labels REGB, RESB, CONB, INTB, and COMLBL.

Note that data with security label COMLBL can be seen by users in either project A or project B.

Users logged on with a particular security label:

- Can *read or update* data with lower security labels in that project, plus COMLBL
- Cannot read data with higher security labels in that project.

For example, users logged on with security label RESA can:

- Read or update data with security label RESA, CONA, INTA, and COMLBL
- Cannot read data with security label REGA.

Note: INTA represents something of a special case. Because the INTA security label is not defined to include the COMLBL category, users who log on with the INTA security label can only access data with the INTA security label.

Figure 8 (Part 3 of 4). Security Labels: An Example

What Users Can Do Based on Security Label Authorization Checking (SETROPTS MLS(FAILURES) in Effect):

Note: Other authorization requirements, such as access lists and UACC, can prevent users from accessing data. This figure is limited to the controls enforced by security label authorization checking when the SECLABEL class is active and the SETROPTS MLS(FAILURES) option is in effect.

Users who log on with project A security labels can only view data with security labels REGA, RESA, CONA, INTA, and COMLBL.

Users who log on with project B security labels can only view data with security labels REGB, RESB, CONB, INTB, and COMLBL.

Note that data with security label COMLBL can be seen by users in either project A or project B.

Users logged on with a particular security label:

- Can update data with that security label
- Can read data with lower security labels in that project, plus COMLBL
- Cannot read data with higher security labels in that project.

For example, users logged on with security label RESA can:

- Update data with security label RESA
- Read data from lower security labels in that project: CONA, INTA, and COMLBL
- Cannot read data with security label REGA.

Note: INTA represents something of a special case. Because the INTA security label is not defined to include the COMLBL category, users who log on with the INTA security label can only access data with the INTA security label.

Figure 8 (Part 4 of 4). Security Labels: An Example

Chapter 8. Creating Resource Group Profiles

Like generic profiles, *resource group profiles* enable you to protect multiple resources with one profile. However, the resources do not have to have similar names.

A resource group profile is a general resource profile with the following special characteristics:

- Its name does not match the resources it protects.
- The ADDMEM operand (not the profile name itself) specifies the resources it protects.
- Its class is a resource group class (for example, GTERMINL).
- The related member class (not the resource group class itself) must be RACLISTed. For example, the TERMINAL class must be RACLISTed, not the GTERMINL class. Depending on the class, RACLISTing is accomplished using the SETROPTS command (TERMINAL class) or the RACROUTE REQUEST=LIST macro (the other classes).

For example, the following profile:

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

Protects three terminals that have *unlike* names:

```
M01RF267
M03RF168
M04GG148
```

Table 14 shows the resource group classes and their related member classes.

Table 14. Resource Group Classes

Resource	**Resource Group Class**	**Related Member Class**
Terminals	GTERMINL	TERMINAL
Installation-defined classes	Installation-defined class names	Installation-defined class names

To use resource group profiles, take the following steps (terminals are used as a readily understood example):

1. Create the resource group profile:

   ```
   RDEFINE  GTERMINL  profile-name  UACC(NONE)
            ADDMEM(resource-name-with-or-without-generic-character...)
   ```

   where:

   GTERMINL is the resource group class for terminals.

   profile-name is a discrete profile name of your choice (generic characters are not allowed).

   resource-name... is the name of the resource to be protected, for example, a terminal ID. If you first activate generic profile checking for the related member class, you can include a generic character (* or % only) in the resource name.

2. Grant the appropriate access to the appropriate users and groups. In the following example, READ access is given to users in group GROUPA:

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

3. When you are ready to start using the protection defined in the profiles, activate the *member class*. You must also activate SETROPTS RACLIST processing for the *member class*.

   For example, for terminals, issue the following command:

   ```
   SETROPTS  CLASSACT(TERMINAL)  RACLIST(TERMINAL)
   ```

   Note: Any time you make a change to a GTERMINL profile, you must also refresh SETROPTS RACLIST processing for the TERMINAL class for the change to take effect:

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

Adding a Resource to a Profile

To add a resource to a profile, issue the RALTER command with the ADDMEM operand, then refresh the in-storage profiles for that class. For example:

```
RALTER  GTERMINL  DEPT35  ADDMEM(M01RF268)
SETROPTS  RACLIST(TERMINAL)  REFRESH
```

Deleting a Resource from a Profile

To delete a resource from a profile, issue the RALTER command with the DELMEM operand, then refresh the in-storage profiles for that class. For example:

```
RALTER  GTERMINL  DEPT35  DELMEM(M01RF268)
SETROPTS  RACLIST(TERMINAL)  REFRESH
```

Which Profiles Protect a Particular Resource?

RACF does not prevent you from specifying the same resource in more than one resource grouping profile. If you do so, more than one profile is used to determine the actual protection used. (See “Which Profile Is Used?.”) It can be difficult to determine exactly what protection any one resource has.

To find out if more than one profile protects a particular resource, issue the RLIST command with the RESGROUP operand as follows:

```
RLIST  TERMINAL  resource-name  RESGROUP
```

Make sure to specify the member class (such as TERMINAL) on the RLIST command. The profiles that protect the terminal will appear in the RLIST output under “RESOURCE GROUPS.”

For example, assume that the following commands were issued:

```
RDEFINE  GTERMINL  DEPT20  ADDMEM(T1 T2 T3)
RDEFINE  GTERMINL  DEPT22  ADDMEM(T3)
```

If you issue the following command:

```
RLIST  TERMINAL  T3  RESGROUP
```

the RLIST output will include the following:

```
RESOURCE GROUPS
-------- ------
DEPT20 DEPT22
```

Note: If a “member class” profile exists for the resource (in this example, if RDEFINE TERMINAL T3 had been issued), the RLIST output includes both the resource groups and the listing of the TERMINAL profile.

Which Profile Is Used?

If a resource is protected by more than one profile, all the profiles are checked and the following is done:

- The most restrictive UACC is used.
- For any particular user, the least restrictive of the access entries is used.
- The highest security level is used.
- Auditing is done if requested by any of the profiles.
- Category lists are combined.
- The first SECLABEL field found is chosen.

Note: The data supplied to the RACROUTE REQUEST=LIST exits contains flags to change the above rules if you desire. For more information, see *RACF System Programmer's Guide*.

Recommendation: Do *not* specify the same resource in more than one profile.

Considerations for Resource Group Profiles

- Do not issue the SETROPTS RACLIST command for the resource group class (for example, GTERMINL). Instead, specify the related member class (for example, TERMINAL). When you RACLIST the TERMINAL class, RACF will RACLIST the GTERMINL class for you.
- You cannot specify generic profile names in the resource group class.
- You *can* specify generic names on the ADDMEM operand; however, you should consider defining your generics in the MEMBER class so that the RLIST command can be used to find which generic profile produces a resource.
- A resource group profile is associated with one and only one resource class and cannot be used to group resources from two different classes.
- If you use resource grouping profiles, you should consider avoiding the use of the related member class. For example, if you use GTERMINL profiles, convert entirely to using GTERMINL profiles, then delete all TERMINAL profiles. This can ease the administration of terminal authorizations. For example, the SEARCH command will list profile names for only one class at a time: GTERMINL or TERMINAL.

  When converting generic TERMINAL profiles to GTERMINL profiles, you can specify generic characters on the ADDMEM operand to obtain the same coverage.

Chapter 9. The RACF Database Unload Utility (IRRDBU00)

The RACF database holds an installation's security data. This data is used to control access to resources, verify users, and generate a variety of reports dealing with system usage and integrity. Standard reports are provided and used to determine if the installation's security objectives are being met.

The RACF database unload utility enables installations to create a sequential file from a restructured RACF database. The sequential file can be used in several ways: viewed directly, used as input for installation-written programs, and manipulated with sort/merge utilities. It can also be uploaded to a database manager (for example, SQL/DS) to process complex inquiries and create installation-tailored reports.

Diagnostic Capability

Although the design rational of the IRRDBU00 utility is not diagnosis, this utility does provide some useful diagnostic capabilities. Since the IRRDBU00 utility must read every profile in the entire RACF database, it provides a side effect of validating profile data. While this is not a comprehensive validation of every field value, it is a validation of many lengths and count fields which are needed to successfully read each profile.

This side effect may be used to help identify a profile in error. If IRRDBU00 encounters a profile in error, it may issue message IRR67092. This message contains an ICHEINTY return and reason code and also the entry name of the profile being processed. If you do not receive this message, but rather abend or terminate in another fashion you may also be able to determine the profile in error. To do this, look in the output data set and find the last profile, (at the bottom), that was unloaded. It is likely that this profile is okay, however; the next profile in the database, (in the same class), is likely to be the culprit if indeed a bad profile is causing the utility to terminate.

See *RACF System Programmer's Guide* and and *RACF Diagnosis Guide* for more information on RACF database diagnosis and correction.

Performance Considerations

IRRDBU00 processes either a copy of the RACF database, a backup RACF database, or the active RACF database. You must have UPDATE authority to the database. It is *recommended* that you run the utility against a recent copy of your restructured RACF database using the NOLOCKINPUT parameter.

As IRRDBU00 executes, it issues one RESERVE per profile, not a continuous RESERVE (this is also the case in IRRUT100 processing). When IRRDBU00 has finished copying a profile, it releases the RESERVE on it. Consider this possible impact to performance if you select your active RACF database as input. Running IRRDBU00 against a *copy* of the database causes the least impact to system performance.

An installation can choose to unload its database with one utility invocation, or if it has split its database, it can unload individual pieces of its database with separate utility invocations. These utility invocations can execute concurrently.

Operational Considerations

The output records of IRRDBU00 are determined by the structure of the RACF database. The utility unloads all profiles in the database. It does not unload all fields in each profile and treats some fields in a special way. Fields containing customer data are unloaded exactly as they appear in the database. Encrypted and reserved fields are not unloaded.

Although the maximum length unloaded for most fields is 255 bytes, all 1023 bytes of data for the FSROOT, HOME, and PROGRAM fields in the user's OVM segment are unloaded.

For the conversion rules of the database unload utility, see *RACF Macros and Interfaces*.

The database unload utility uses the class descriptor tables (IBM-supplied and installation-defined) as it unloads profiles. If your database is imported from another system, you may also have to import the class descriptor tables (ICHRRCDX and ICHRRCDE). Classes will be unloaded only if there is an entry for them in ICHRRCDE or ICHRRCDX on the system running the utility.

To correlate the RACF profiles with the data unloaded by the utility, see *RACF Macros and Interfaces*.

Using IRRDBU00 on VM (RACFDBU)

VM installations use the RACFDBU EXEC to execute the database unload utility. For installations that share a RACF database between MVS and VM, the recommendation is to execute the unload utility from the MVS environment.

You can execute the IRRDBU00 utility either by panel invocation or command invocation. For details, see “Panel Invocation of RACFDBU” on page 136 and “Command Invocation of RACFDBU” on page 138.

RACFDBU Setup

Before unloading your restructured RACF database, you must:

1. Log on to a virtual machine that has links to the RACF service machine's 490 disk and the RACF database disks (200, 300, and others).
2. Have at least READ access to the database to unload if using parameter NOLOCKINPUT.

   You must have WRITE access to the database to unload if using parameters UNLOCKINPUT or LOCKINPUT.

   For more information, see “Allowable Parameters” on page 139.
3. Ensure that the database to unload is restructured (block size is 4096).
4. IPL the RACF service machine's 490 disk.
5. Ensure that there is adequate free space on the output minidisk to contain the utility output file.

   The size of the output file is roughly estimated as twice the size of the used portion of the input database, but you must also consider the type of profiles in your database. For example, profiles having variable length fields, such as installation data, require more space when unloaded because the maximum size of the field is unloaded (up to 255 bytes for most fields).

   Although the maximum length unloaded for most fields is 255 bytes, all 1023 bytes of data for the FSROOT, HOME, and PROGRAM fields in the user's OVM segment are unloaded.

   Determine the percentage of space your database is using by running the RACUT200 EXEC and use that percentage to guide you in allocating the output file. For example, if your database has 100 cylinders allocated and you are using 35% of it, you will need approximately 70 free cylinders on your output minidisk.

Note: On a VM system, there is no ability to span DASD packs; the size of minidisk output is restricted to one physical DASD device (minidisk).

6. Link the output minidisk as R/W.

Split Database Considerations

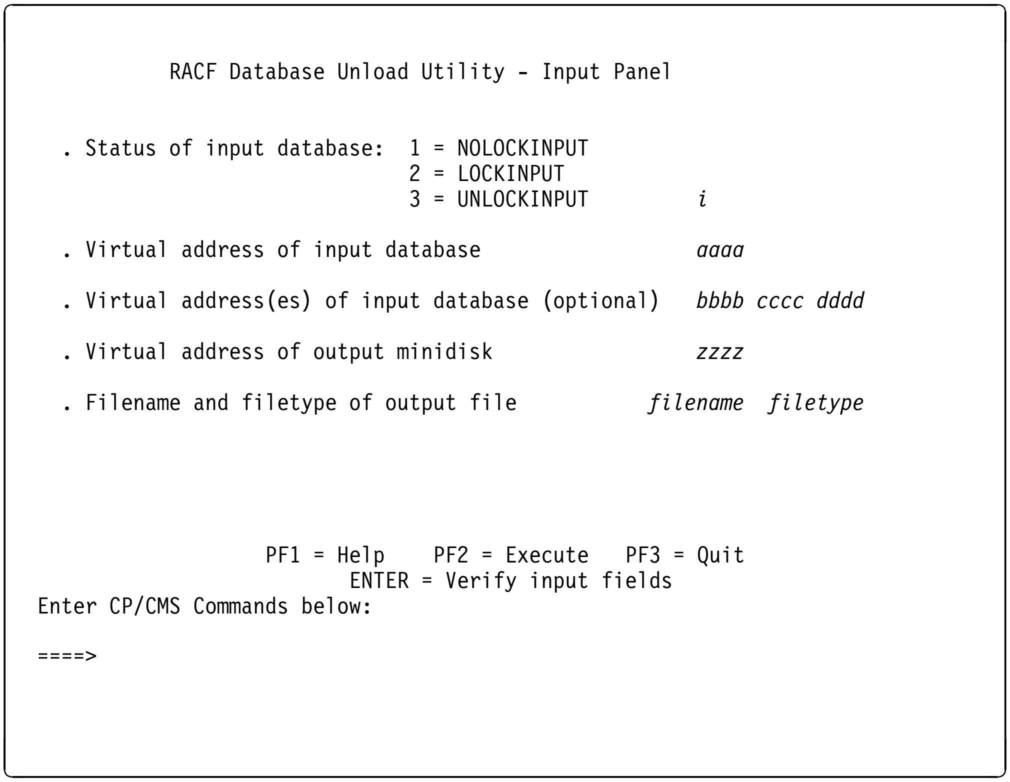
If your database has been split into several parts (maximum of 4), you may unload each part with separate utility invocations or you may unload all the parts with one utility invocation.

To unload all the parts of a split database with one utility invocation, you must enter the virtual addresses of *all* the database parts in your command or panel invocation. The order of the virtual addresses entered for the input databases must match the virtual addresses specified in the RACONFIG EXEC for the database parts.

To unload only one part of a split database with a utility invocation, you must enter the virtual addresses of the specific database part.

Panel Invocation of RACFDBU

Begin the exec by entering `RACFDBU` on the command line. The input panel appears on your screen. Figure 9 illustrates the RACFDBU input panel.



```
          RACF Database Unload Utility - Input Panel


  . Status of input database:  1 = NOLOCKINPUT
                               2 = LOCKINPUT
                               3 = UNLOCKINPUT        i

  . Virtual address of input database                aaaa

  . Virtual address(es) of input database (optional)  bbbb cccc dddd

  . Virtual address of output minidisk               zzzz

  . Filename and filetype of output file         filename  filetype




                  PF1 = Help    PF2 = Execute   PF3 = Quit
                         ENTER = Verify input fields
Enter CP/CMS Commands below:

====>
```

Figure 9. Input Panel for RACFDBU

The user must supply the following values:

i The status of the input database RACFDBU is processing.

For NOLOCKINPUT, enter the numeric 1.
For LOCKINPUT, enter the numeric 2.
For UNLOCKINPUT, enter the numeric 3.

See “Allowable Parameters” on page 139 for an explanation of these parameters. If UNLOCKINPUT is selected, it is not necessary to provide an output minidisk. UNLOCKINPUT unlocks the input database.

This field is required; 1 (NOLOCKINPUT) is the default.

aaaa The virtual address of the input RACF database to unload. The blocksize of the input database must equal 4096. Using LOCKINPUT or UNLOCKINPUT on the database requires the user to have R/W access to this disk.

Note: If the input database is on FBA DASD that contains more than one file, the user will be prompted for the filename and filetype of the database to unload.

This is a required field.

bbbb cccc dddd
The virtual addresses of the input database to unload if the database has been split into parts. The block size of these input database parts must be 4096.

Note: FBA DASD does not support split databases, and if these fields are specified, they will be ignored.

These are optional input fields.

zzzz The virtual address of a minidisk where the output from RACFDBU will be written.

This is a required field.

filename filetype
The file name and file type of the output file. RACFDBU OUTPUT is the default. You can supply another file name or file type.

If the output file you specify already exists, the utility changes the file type name of the existing file. For example, if the default file (RACFDBU OUTPUT) exists on the output minidisk, the existing file is copied to a file named RACFDBU OUTPUT1 on the same disk. It overlays any previous RACFDBU OUTPUT1 file. If the file type is 8 characters long, the last character is changed to a 1.

The input values entered on the panel are saved and reappear the next time you invoke RACFDBU. After you have entered your input in the required fields, press one of the following keys. The meaning of the ENTER key and the PF key definitions are:

Key	**Meaning**
Enter	Verify user input. Messages are issued if there are errors. (The database unload utility is not invoked.)
PF1	Display help screen.
PF2	Execute key. All input fields are validated and, if valid, the unload utility is invoked.

PF3 Exit panel.

Command Invocation of RACFDBU

Your installation may want to run the IRRDBU00 utility without interactive processing. To start the utility automatically, you can invoke the utility from a command line or a user-written exec, but input parameters must be correctly specified. The command invocation fields are similar to the panel invocation fields.

Note: If you are using FBA DASD and have more than one file on the minidisk, you must supply the file name and file type of the input RACF database.

All required parameters must be valid or the database unload utility will not be invoked.

The virtual addresses of an input RACF database and an output minidisk are required. There are defaults for ISTATUS, OUTFN, and OUTFT. If your database is on FBA DASD and there is more than one file on the minidisk, you must supply the file name and file type of the RACF database.

Syntax for command invocation

```
RACFDBU aaaa [bbbb [cccc [dddd ]]] zzzz [(options...[)]]

Options:
[OUTFN filename]
[OUTFT filetype]
[ISTATUS i]
[FBAFN filename FBAFT filetype]
```

The explanation of the input fields follows:

aaaa The virtual address of the input RACF database to unload. The block size of the input database must equal 4096. Using LOCKINPUT or UNLOCKINPUT on the database requires the user to have R/W access to this disk.

Note: If the input database is on FBA DASD that contains more than one file, the user must provide FBAFN and FBAFT.

This is a required field.

bbbb cccc dddd
The virtual addresses of the input database to unload if the database has been split into parts. The blocksize of these input database parts must be 4096.

Note: FBA DASD does not support split databases, and if these fields are specified, they will be ignored.

zzzz Virtual address of output R/W minidisk.

This is a required field.

OUTFN *filename*
Filename of output CMS file.

This field is optional. RACFDBU is the default.

OUTFT *filetype*
Filetype of output CMS file.

This field is optional. OUTPUT is the default.

ISTATUS *i*
The status of the input database RACFDBU is processing.

For NOLOCKINPUT, enter the numeric 1.
For LOCKINPUT, enter the numeric 2.
For UNLOCKINPUT, enter the numeric 3.

See "Allowable Parameters" for an explanation of these parameters.

This field is optional. 1 (NOLOCKINPUT) is the default.

FBAFN *filename*
Filename of RACF database on FBA DASD.

The field is required if there is more than one file on the input minidisk.

FBAFT *filetype*
Filetype of RACF database on FBA DASD.

The field is required if there is more than one file on the input minidisk.

Command Invocation Return Codes

To determine if the database unload utility successfully executed, check the return code. A return code of 0 indicates successful utility execution. A return code of 16 indicates that the utility did not execute. It is issued with error messages indicating the reason for failure.

IRRDBU00 Utility Messages on VM

Messages issued by the IRRDBU00 utility (IRR67xxx messages) are placed in a file named RACFDBU MESSAGES on the user's A-disk. The IRR67xxx messages are documented in *RACF Messages and Codes*.

Messages from RACFDBU appear on the input screen and are documented in *RACF Messages and Codes*. Messages issued by the RACFDBU EXEC begin with RPIDBU.

Allowable Parameters

One of the following parameters must be specified when running the database unload utility. On VM, indicate the parameter numerically: 1 for NOLOCKINPUT, 2 for LOCKINPUT, and 3 for UNLOCKINPUT.

For the *least* impact to system performance, use a copy of your restructured RACF database as input and specify the NOLOCKINPUT parameter.

Using the backup copy of the RACF database is allowed, as long as the backup is in the restructured format. Using an active copy of the restructured RACF database can impact system performance and it is not recommended.

- NOLOCKINPUT

This allows the unload to be performed and does not change the state of the input database. If the database is locked, it remains locked. If it is unlocked, it remains unlocked.

For the least impact to system performance, use a copy of your restructured RACF database as input and specify the NOLOCKINPUT parameter.

Attention

If you use NOLOCKINPUT on the *active* database, your unloaded database may contain inconsistencies.

- LOCKINPUT

  This ensures that the RACF database used as input is not updated by other jobs while the utility is running.

  Note: Statistics are updated.

  If you are running against an active RACF database, LOCKINPUT is recommended.

  Specifying LOCKINPUT means updates are no longer allowed to an input database until the utility terminates.

  If you run IRRDBU00 and use LOCKINPUT, any activity updating the RACF database (such as users logging on and changing passwords and batch jobs allocating new data sets requiring the creation of RACF profiles) will fail with either an ABEND483 RC50 or ABEND485 RC50.

  If using LOCKINPUT, do not schedule maintenance spanning midnight. If the RACF database remains locked past midnight, no new jobs will be able to be submitted, nor will users be able to log on, unless you disable the gathering of logon statistics by issuing a SETROPTS NOINITSTATS command. All steps that require a locked database must be performed on the same calendar day.

  The database unload utility *unlocks* the RACF database after processing with LOCKINPUT specified if the database was unlocked when the utility started. The unload utility output is for report generation and does not replace the input database, which is your primary, active, RACF database.

  This is different from the IRRUT400 and IRRDSC00 utilities, which keep the input database locked and create new output databases. This is done to maintain integrity between the input database and the output database. IRRUT400 and IRRDSC00 assume that you no longer want or need the input database.

- UNLOCKINPUT

  UNLOCKINPUT is used to unlock a database that had been previously locked by the LOCKINPUT keyword. This action enables your input database and allows it to be updated.

  No data unloading is done when this parameter is used.

IRRDBU00 Output

The database unload utility reads every profile in a restructured RACF database. Depending on the contents of the profile, records of various types are created. All the records are written to a sequential file. For details on the format of the records created, see *RACF Macros and Interfaces*.

This sequential output file can be:

- Used as input to a database management system, such as DB2 or SQL/DS.
- Manipulated with sort/merge utilities
- Used as input to your own programs
- Viewed directly

Sample SQL mappings are provided for VM installations, and the names of the files are:

- RACDBUTB SAMPLE
- RACDBULD SAMPLE
- RACDBUQR SAMPLE

See “Using the Database Unload Utility Output with SQL/DS” on page 141 for examples of using the IRRDBU00 output with SQL/DS.

Using the Database Unload Utility Output

The output file from the database unload utility can be:

- Viewed directly.
- Used as input to your own programs.
- Manipulated with sort/merge utilities.
- Used as input to a database management system. Installations will be able to produce reports tailored to their requirements.

Sort/Merge Programs

The database unload utility processes all the profiles in the input database. If you want a subset of the output records, you can use a standard utility such as DFSORT/CMS to select them.

Relational Databases

Much of the function of the database unload utility is not realized until the data it creates is loaded into a relational database management system (DBMS) such as SQL/DS.

Using the Database Unload Utility Output with SQL/DS

The records produced by the database unload utility are designed to be processed by the SQL/DS Load Utility or its equivalent. The definition and control statements for a SQL/DS utilization of the output are as follows:

- Sample data definition language (DDL) statements to define the relational representation of the RACF database and sample SQL/DS definitions which perform database and index creation. These are in file RACDBUTB SAMPLE.

- Sample control statements for the SQL/DS Load Utility that map the output from the database unload utility (IRRDBU00). These are in file RACDBULD SAMPLE.
- Sample Structured Query Language (SQL) queries that perform useful data inquiries:
  - A query to list all the members of a group
  - A query to list all the users with the global-SPECIAL attribute
  - A query to find all the groups a user is connected to
  - A query to find all the data set access lists containing user IDs no longer valid.

  These are in file RACDBUQR SAMPLE.

  Note: The files RACDBUTB SAMPLE, RACDBULD SAMPLE, and RACDBUQR SAMPLE are shipped with RACF on the RACF service machine 305 disk.

For complete information on SQL/DS, see:

- *SQL/Data System General Information for IBM VM Systems*
- *SQL/Data System Database Administration for IBM VM Systems*
- *SQL/Data System System Administration for IBM VM Systems*
- *SQL/Data System SQL Reference for IBM VM Systems and VSE*

Steps for Using IRRDBU00 Output with SQL/DS

To create and manage a SQL/DS database containing the database unload utility output, you must:

1. Create one or more SQL/DS DBSPACEs.
2. Create SQL/DS tables.
3. Create the SQL/DS indexes.
4. Load data into the tables.
5. Reorganize the indexes (optional).
6. Delete table data (optional).

The first three steps are initial setup, and you can choose to run them once. When you get new data to import into the SQL/DS database, you erase your current table data. You then reload and reorganize your indexes.

The following sections show examples of the SQL/DS utility input for these functions.

Creating a SQL/DS DBSPACE

SQL/DS stores tables and indexes on tables in DBSPACEs. A DBSPACE is a logical allocation of space in the database. For more information see *SQL/DS System Administration*.

Creating the SQL/DS Tables

After the DBSPACE is created, SQL statements that define the tables are executed. Figure 10 contains an example of the SQL statements required to create a table for the Group Basic Data record of the database unload utility.

The RACDBUTB SAMPLE file contains examples that create separate tables for each record type produced by the database unload utility. The user must supply the user ID (*userid*).

```
CREATE TABLE userid.GROUP_BD (
     GPBD_NAME          CHAR(8)    NOT NULL,
     GPBD_SUPGRP_ID     CHAR(8),
     GPBD_CREATE_DATE   DATE,
     GPBD_OWNER_ID      CHAR(8)    NOT NULL,
     GPBD_UACC          CHAR(8)    NOT NULL,
     GPBD_NOTERMUACC    CHAR(1)    NOT NULL,
     GPBD_INSTALL_DATA  CHAR(254),
     GPBD_MODEL         CHAR(44)
     )
     IN GROUP_BD.
     ;
```

Figure 10. Sample SQL Utility Statements Creating a Table

Creating the SQL/DS Indexes

SQL/DS performance improves with the use of indexes. The RACDBUTB SAMPLE file creates an index for every primary key and every foreign key identified in the record types. Figure 11 contains sample statements to create the indexes for the Group Basic Data record.

```
CREATE UNIQUE INDEX userid.GROUP_BD_IX1
       ON userid.GROUP_BD
          (GPBD_NAME)
          ;

CREATE        INDEX userid.GROUP_BD_IX2
       ON userid.GROUP_BD
          (GPBD_NAME, GPBD_SUPGRP_ID)
          ;

CREATE        INDEX userid.GROUP_BD_IX3
       ON userid.GROUP_BD
          (GPBD_OWNER_ID)
          ;

CREATE        INDEX userid.GROUP_BD_IX4
       ON userid.GROUP_BD
          (GPBD_MODEL)
          ;
```

Figure 11. Sample SQL Utility Statements Creating Indexes

Loading the SQL/DS Tables

Figure 12 shows the statements required to load the Group Basic Data record. The RACDBULD SAMPLE file contains statements that load all the record types produced by the database unload utility. The sample requires that the output of RACFDBU be made into a fixed record length file.

```
DATALOAD TABLE (GROUP_BD)           IF POS(1:4)='0100'
          GPBD_NAME                  006-013
          GPBD_SUPGRP_ID             015-022
          GPBD_CREATE_DATE           024-033
          GPBD_OWNER_ID              035-042
          GPBD_UACC                  044-051
          GPBD_NOTERMUACC            053-053
          GPBD_INSTALL_DATA          058-311
          GPBD_MODEL                 314-357
          INFILE(IRRDBU00)
          )
```

Figure 12. SQL/DS Utility Statements Required to Load the Tables

Note: You can choose not to load some of the tables.

Reorganizing the Indexes in the SQL/DS Database

Queries are processed faster if they are performed against an organized database. SQL/DS provides a utility that allows you to reorganize the indexes on the catalog tables. For more information, see *SQL/DS Database Administration for VM*.

Deleting Data from the SQL/DS Database

Before you reload the database with new data, you should delete the old data. This can be done in several ways:

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

   Figure 13 shows the sample SQL statements that delete the group record data from the tables.

```
           DELETE FROM userid.GROUP_BD           ;
           DELETE FROM userid.GROUP_DFP_DATA     ;
           DELETE FROM userid.GROUP_INSTALL_DATA ;
           DELETE FROM userid.GROUP_SUBGROUPS    ;
           DELETE FROM userid.GROUP_MEMBERS      ;
```

Figure 13. SQL Utility Statements Required to Delete the Group Records

SQL/DS Table Names

The RACDBUTB SAMPLE file creates SQL/DS tables for each record type. Table 15 on page 145 provides a useful reference of record type, record name, and SQL/DS table name.

Table 15. Correlation of Record Type, Record Name, and SQL/DS Table Name

Record Type	Record Name	SQL/DS Table Name
0100	Group Basic Data	GROUP_BD
0101	Group Subgroups	GROUP_SUBGROUPS
0102	Group Members	GROUP_MEMBERS
0103	Group Installation Data	GROUP_INSTALL_DATA
0110	Group DFP Data	GROUP_DFP_DATA
0130	Group OVM Data	GROUP_OVM_DATA
0200	User Basic Data	USER_BD
0201	User Categories	USER_CATEGORIES
0202	User Classes	USER_CLASSES
0203	User Group Connections	USER_GROUPS
0204	User Installation Data	USER_INSTALL_DATA
0205	User Connect Data	USER_CONNECT_DATA
0210	User DFP Data	USER_DFP_DATA
0220	User TSO Data	USER_TSO_DATA
0230	User CICS Data	USER_CICS_DATA
0231	User CICS Operation Classes	USER_CICS_OPCLASS
0240	User Language Data	USER_LANGUAGE_DATA
0250	User OPERPARM Data	USER_OPERPARM_DATA
0251	User OPERPARM Scope	USER_OPERPARM_SCOP
0260	User WORKATTR Data	USER_WORKATTR_DATA
02A0	User OVM Data	USER_OVM_DATA
0400	Data Set Basic Data	DS_BD
0401	Data Set Categories	DS_CATEGORIES
0402	Data Set Conditional Access	DS_COND_ACCESS
0403	Data Set Volumes	DS_VOLUMES
0404	Data Set Access	DS_ACCESS
0405	Data Set Installation Data	DS_INSTALL_DATA
0410	Data Set DFP Data	DS_DFP_DATA
0500	General Resource Basic Data	GENR_BD
0501	General Resource Tape Volumes	GENR_TAPE_VOLUMES
0502	General Resource Categories	GENR_CATEGORIES
0503	General Resource Members	GENR_MEMBERS
0504	General Resource Volumes	GENR_VOLUMES
0505	General Resource Access	GENR_ACCESS
0506	General Resource Installation Data	GENR_INSTALL_DATA
0507	General Resource Conditional Access	GENR_COND_ACCESS
0510	General Resource Session Data	GENR_SESSION_DATA
0511	General Resource Session Entities	GENR_SESSION_ENT
0520	General Resource DLF Data	GENR_DLF_DATA
0521	General Resource DLF Job Names	GENR_DLF_JOBNAMES

Samples Using the Database Unload Utility Output

A relational database management system such as DB2 or SQL/DS can be used with the Query Management Facility (QMF) to create reports.

A report many installations find useful is a list of all data set profiles containing an invalid ID in the access list. This situation occurs when a security administrator deletes a user ID or group ID without deleting the authorities the ID may have had in the RACF database.

To search the data set access list for user IDs and group IDs that do not have user or group profiles in the database, perform the following steps:

1. Create a query that compares the entries in the access list with a list of valid user IDs or group IDs. A sample SQL query is provided in Figure 14 on page 146.
2. Format the results of the query as provided by the QMF form in Figure 15 on page 147.

The resulting report is shown in Figure 16 on page 147.

Note: If you used the IRRUT100 utility to check the references to a user or group ID, IRRUT100 requires that the user or group ID be known. The sample query does not have such a requirement. It finds *all* user IDs or group IDs that are not valid.

When your RACF database was unloaded, the IRRDBU00 utility created a Data Set Access Record (record type 404) for each user ID or group ID in the access list of each data set.

When you loaded your IRRDBU00 output into DB2 or SQL/DS, an AUTH_IDS table was created that contains the name of every valid user ID and group ID.

SQL Query

The sample SQL query compares the ID in the data set access record (DSACC_AUTH_ID) with the list of valid user and group IDs (in AUTH_IDS). When a user ID is found that is not a valid user ID or group ID, it is listed. The query also lists the data set profile name, the authority that the user has, and the access count.

```
------------------------------------------------------------------------
-- Description: Check all of the data set standard access lists and   --
--              verify that each user ID is a valid user or           --
--              group ID                                              --
--                                                                    --
-- Tables Accessed: SQL                                               --
--                  "DS_ACCESS"     - A list of dataset authorities   --
--                  "AUTH_IDS"      - A list of valid user/group IDs  --
--                                                                    --
------------------------------------------------------------------------
          SELECT
                 DSACC_NAME
                ,DSACC_AUTH_ID
                ,DSACC_ACCESS
                ,DSACC_ACCESS_CNT
          FROM
                USER01.DS_ACCESS X
          WHERE NOT EXISTS
                ( SELECT *
                   FROM
                        USER01.AUTH_IDS
                   WHERE
                   X.DSACC_AUTH_ID=AUTHID_NAME
                     )
            AND
                   X.DSACC_AUTH_ID¬='*'
            ORDER BY 1
            ;
```

Figure 14. A Sample SQL Query

QMF Form

If the SQL query shown in Figure 14 is processed using QMF, the data that is returned can be processed into a report. Figure 15 shows a report or forms definition. It creates the report shown in Figure 16 entitled “Data Set Profiles With Users Who Are Not Valid in the Access List.”

```
COLUMNS:                Total Width of Report Columns: 80
 NUM COLUMN HEADING                         USAGE  INDENT WIDTH EDIT SEQ
 --- -------------------------------------- ------ ------ ----- ---- ---
   1 DSACC_NAME                             BREAK1 2      44    C    1
   2 DSACC_AUTH_ID                                 2      8     C    2
   3 DSACC_ACCESS                                  2      9     C    3
   4 DSACC_ACCESS_CNT                              2      11    L    4



 PAGE:    HEADING  ===> DATA SET PROFILES WITH USERS WHO ARE NOT VALID IN THE ACCESS LIST
          FOOTING  ===>
 FINAL:   TEXT     ===>
 BREAK1:  NEW PAGE FOR BREAK? ===> NO
          FOOTING  ===>
 BREAK2:  NEW PAGE FOR BREAK? ===> NO
          FOOTING  ===>
 OPTIONS: OUTLINE? ===> YES                   DEFAULT BREAK TEXT? ===> NO
```

Figure 15. A Sample QMF Form

Report Output

This is the report that results from the SQL query shown in Figure 14 and the QMF form shown in Figure 15. Not all resulting rows are shown.

```
  DATA SET PROFILES WITH USERS WHO ARE NOT VALID IN THE ACCESS LIST


  DSACC_NAME                        DSACC_AUT DSACC_ACC  DSACC_ACCES
  --------------------------------  --------  ---------  -----------
  MARKN.WORK.CNTL                   WAYNEN    READ                 3
                                    DONE      READ                 2
                                    DANJ      READ                 2
                                    ANNEL     READ                 4
                                    TOMB      READ                 5
                                    ISABELH   READ                 2
                                    BOBS      READ                 1
                                    SUSANB    READ                 2
                                    DANL      READ                 3
                                    GLENS     READ                 3
                                    LOUL      READ                 6

  SUSANL.DOC.TEXT                   BOBS      READ                 4
                                    DANL      READ                 2

  TAMMY.TEST.DATA                   GLENS     READ                10

  WALT.DESIGN.DATA                  LOUL      READ                 3

  LAURIE.*                          DONE      UPDATE               6
                                    TOMB      READ                 1



05/31/1995 04:26 PM                                            PAGE 10
```

Figure 16. A Sample Report

Chapter 10. Using VMXEVENT Profiles to Protect CP Resources

When RACF is installed, the VM operating system calls RACF to provide security protection beyond the operating system's basic features. This chapter discusses how you can use RACF to meet your operating system security needs. Previous chapters provided an overview to assist you in understanding your security needs and defining an installation security policy. Once you understand your security policy, use this chapter to begin implementing resource protection.

Resource protection involves two decisions: what resources an installation wants to protect and the type of authorization required to access the resources. The general resource classes that RACF provides can be used to protect minidisks, terminals, restricted segments, certain diagnose codes and CP commands, RSCS nodes, and virtual unit record devices. The general resource classes for VM are described in "Protecting General Resources" on page 20.

The security administrator implements the installation's security policy by using RACF commands to:

- Define profiles in a general resource class
- Permit users to the resources within the class
- Activate the class

For a complete discussion of the general resource classes and how they can be used to protect your VM resources, see "Protecting VM Resources" on page 163.

The VM operating system determines the VM events that require authorization checking by RACF. These are referred to as *controllable* VM events. By default, control is on for certain VM events. The security administrator determines whether the default controls are needed for implementing the installation's security policy as control can be turned off by using profiles in the VMXEVENT class.

For a complete discussion of the VMXEVENT class and controllable events, see "Controlling Authorization Checking for VM Events" on page 152.

The VM operating system also provides auditable VM events. By default, there is no auditing done on any VM events. For a complete discussion on auditing, see *RACF Auditor's Guide*.

Table 16 on page 152 shows the relationship of resources to protect, the VM name of the resource (VM event), and the RACF class associated with the resource. In some cases, there is not a one-to-one relationship.

Table 16. VM Events, Resources, and RACF Classes

VM Event	Resource to Protect	RACF Class
APPCPWVL	APPC connection with password	USER
AUTOLOG	System access	USER
COUPLE.G	Connection to a Guest LAN	VMLAN
DIAG0A0	Use of Diagnose Code X'A0'	VMCMD
DIAG0D4	Alternate User ID	VMBATCH
DIAG0E4	Use of Diagnose Code X'E4'	VMCMD
DIAG280	BFS set-UID and set-GID files	VMPOSIX
LINK	Links to other's minidisks	VMMDISK
LOGON	System access	USER
	Terminals	TERMINAL GTERMINL
MDISK	Links to own minidisks	VMMDISK
RSTDSEG	Restricted segments (NSS or DCSS)	VMSEGMT
STORE.C	Use of STORE HOST command	VMCMD
TAG	RSCS nodes	VMNODE
TRANSFER.D	Virtual unit record devices	VMRDR
TRANSFER.G	Virtual unit record devices	VMRDR
TRSOURCE	Use of the TRSOURCE command	VMCMD
XAUTOLOG (AB version)	System access	USER
XAUTOLOG (G version)	System access	USER
	Ability to XAUTOLOG someone else	VMCMD

For more information about BFS set-UID and set-GID files, see “Protecting Set-UID and Set-GID Executable Files” on page 253.

Controlling Authorization Checking for VM Events

The VM operating system always calls RACF for authorization checking of the LOGON, AUTOLOG and XAUTOLOG events. You cannot turn control off for the LOGON, AUTOLOG and XAUTOLOG events.

The VM operating system may call RACF for authorization checking of certain other VM events. Your security policy may not find these calls necessary. You can use RACF commands to turn control on or off for these VM events.

Controlling these VM events can be done on a system-wide basis using system VM event profiles or for an individual user using individual VM event profiles. Defining these profiles is described in “System VM Event Profile” on page 154 and “Individual VM Event Profile” on page 155.

For z/VM, the following events are controlled, by default:

APPCPWVL
COUPLE.G

DIAG0A0
DIAG0D4
DIAG0E4
DIAG280
LINK
MDISK
RSTDSEG
STORE.C
TAG
TRANSFER.D
TRANSFER.G
TRSOURCE

These events, along with their control settings, are listed in the SETEVENT LIST output. If your installation changed the default settings, the changes will be reflected in the SETEVENT LIST output. Sample SETEVENT LIST output is in Figure 17 on page 162.

If control for an event is off, RACF does not perform authorization checking for the general resource class because VM does not call RACF. If control for an event is on, RACF performs authorization checking for the general resource class relevant to the particular event.

For example, LINK is protected by profiles in the general resource class VMMDISK. LINK is also a controllable VM event. If the VMXEVENT profile indicates authorization checking by RACF (control is on), then the VMMDISK profiles determine if authorization is granted. If the VMXEVENT profile indicates no authorization checking by RACF (control is off), then VM security determines if authorization is granted.

Creating VMXEVENT Profiles

If the default VM event settings do not meet your security needs, you must create a VMXEVENT profile for your installation.

When creating VMXEVENT profiles, remember that they serve a dual purpose. They can be used to instruct VM to call RACF to perform *access checking* on designated VM events and they can be used to instruct VM to call RACF to perform *auditing* on designated VM events.

You can use one profile to define both VM auditing and access calls to RACF. Alternatively, you can use one profile to indicate that VM should call RACF to audit certain events and another profile to indicate that VM should call RACF to perform access checking on certain events.

The ability to create these various types of profiles depends on the attribute of the user creating the profile:

- A user with the SPECIAL attribute can define profiles in the VMXEVENT class, but can *only* set the *control* options for VM events in that profile.
- A user with the AUDITOR attribute and CLAUTH to the VMXEVENT class can define profiles in the VMXEVENT class, but can *only* set the *audit* options for VM events in that profile.

If the SPECIAL user wants the profile to also contain VM events to be audited, the SPECIAL user must place the user with the AUDITOR attribute on the access list of the VMXEVENT profile with an access of ALTER.

To easily adapt to changes in auditing or access control requirements, you might want to define several VMXEVENT profiles to use as your auditing or access control environment changes.

To activate the VMXEVENT resource class, issue the following command:

```
SETROPTS CLASSACT(VMXEVENT)
```

You are now ready to define your profiles.

System VM Event Profile

A system VM event profile is a resource profile defined in the VMXEVENT class. The options set in a system VM event profile determine the type of auditing and control that will take place for all of the users on the system. The rest of this section discusses how to use a system VM event profile to control and not control VM events. For information specific to auditing VM events, see *RACF Auditor's Guide*.

If an individual VM event profile is present for any specific user, it takes precedence over a system VM event profile (see “Individual VM Event Profile” on page 155).

Creating a System VM Event Profile

Use the RDEFINE command to create a system VM event profile. For example, the following command defines a VMXEVENT profile called EVENTS1.

```
RDEFINE VMXEVENT EVENTS1
```

Altering a System VM Event Profile to Stop RACF Authorization Checking

When RACF is first installed, the default is that all controllable events are controlled. This means that VM always calls RACF for authorization checking for these events. If you want RACF to perform authorization checking for these events, you do not need to make any changes. You can use the RALTER command to add a member to a VMXEVENT profile for each event for which you wish to turn off control.

For example, the following command specifies that, when profile EVENTS1 is in effect on the system, VM does not call RACF for authorization checking for the LINK and TAG commands.

```
RALTER VMXEVENT EVENTS1 ADDMEM(LINK/NOCTL TAG/NOCTL)
```

The first part of the member name (VM event) must match exactly the “VM EVENT” shown in the “CONTROLLABLE VM EVENTS” section of the output of the SETEVENT LIST command. For a sample of this output, see Figure 17 on page 162.

You can issue the RALTER command as many times as you need for one VMXEVENT profile, adding or deleting members as necessary.

Notes:

1. You can combine creating and altering the profile by specifying the ADDMEM operand on the RDEFINE command.
2. If you use the RACF ISPF panels to update a VMXEVENT profile, you can select VM event names from a list on the panel.
3. When you issue the RLIST command for a VMXEVENT profile, the output shows the members that have been added to the profile.

Activating a System VM Event Profile

Use the SETEVENT command to specify which VMXEVENT profile you want active. Depending on the access control requirements of your environment at a given time, various profiles are appropriate to meet those requirements. For example, the following command activates the access checking set in the profile EVENTS1.

```
SETEVENT REFRESH EVENTS1
```

When a profile is active, any changes you make to it do not take effect until a subsequent SETEVENT REFRESH command is issued.

Note: If you are sharing a database between a VM/SP system and a z/VM system, you must first issue SETEVENT REFRESH separately on each system on which you wish to audit or control VM events. If you are sharing a database between a VM/XA system and a VM/ESA system or sharing between two or more z/VM systems, you only need to issue the SETEVENT REFRESH command on one system.

Removing VM Events from a System VM Event Profile

Use the RALTER command to remove a member from a profile for each event that is to be returned to its default setting. For example, the following command deletes the settings for the LINK and TAG events which were created in a previous example.

```
RALTER VMXEVENT EVENTS1 DELMEM(LINK/NOCTL TAG/NOCTL)
```

The first part of the member name (VM event) must match exactly the “VM EVENT” shown in the “CONTROLLABLE VM EVENTS” section of the output of the SETEVENT LIST command. For a sample of this output, see Figure 17 on page 162.

For these changes to take effect, the EVENTS1 profile must be reactivated by the SETEVENT REFRESH command, as described in “Activating a System VM Event Profile.” Once the SETEVENT REFRESH has been issued, the settings for the LINK and TAG events will return to the default of being controlled.

Individual VM Event Profile

An individual VM event profile is a resource profile defined in the VMXEVENT class. The options set in an individual VM event profile determine the type of auditing and control that will take place for the user. If present, an individual VM event profile takes precedence over a system VM event profile in determining when VM calls RACF.

The main objective in using an individual VM event profile is to identify users on the system who have unique circumstances in regard to auditing and access control, and to tailor selective profiles to monitor them in a specific way, which may result in either more or less monitoring for these users than for other users on the system.

Suppose, for instance, that a user issues a particular command or series of commands frequently throughout the day. The installation could turn off RACF security checking for these commands by creating an individual VM event profile for the user. As a result of reduced RACF calls, performance may be improved.

Creating an Individual VM Event Profile

Use the RDEFINE command to create an individual VM event profile. Individual VM event profiles are distinguished from system VM event profiles by a high-level qualifier called USERSEL. To identify the profile you are creating as an individual VM event profile, you must specify the high-level qualifier, followed by the user's user ID.

For example, if you wanted to create an individual VM event profile for a user with the user ID of FRANK, enter the following command:

```
RDEFINE VMXEVENT USERSEL.FRANK
```

Notes:

1. It is possible to define an individual VM event profile with no auditing specified, and no events controlled, although you should be aware of the implications. See “Making a User Exempt” on page 158 for more information.
2. Only one individual VM event profile is allowed to be defined for each user; therefore, both control and auditing options must be specified in the same profile.

The profile is put into effect the next time the user ID is logged on, is autologged, or reconnects. To activate the profile while the user ID is logged on, use the SETEVENT command as follows:

```
SETEVENT REFRESH USERSEL.FRANK
```

Altering an Individual VM Event Profile to Stop Authorization Checking

When an individual VM event profile containing no members is activated, the default is that all controllable events are controlled for the user and no events are audited. (To see how to activate auditing for VM events for an individual user, see *RACF Auditor's Guide*.) Note that this is true regardless of what is specified in the system VM event profile. If you do not want RACF to perform authorization checking for specific events for the user, you can use the RALTER command to add a member to the individual VM event profile for each event for which you wish to turn off control.

For example, the following command specifies that when profile USERSEL.FRANK is in effect for user FRANK, VM does not call RACF each time user ID FRANK issues the TAG and the LINK commands.

```
RALTER VMXEVENT USERSEL.FRANK ADDMEM(TAG/NOCTL LINK/NOCTL)
```

The first part of the member name (VM event) must match exactly the “VM EVENT” shown in the “CONTROLLABLE VM EVENTS” section of the output of the SETEVENT LIST command. See Figure 17 on page 162.

You can issue the RALTER command as many times as you need for one VMXEVENT profile, adding or deleting members as necessary.

Notes:

1. You can combine creating and altering the profile by specifying the ADDMEM operand on the RDEFINE command.
2. If you use the RACF ISPF panels to update a VMXEVENT profile, you can select VM events from a list on the panel.
3. The options set in this profile do not take effect until SETEVENT REFRESH USERSEL.FRANK has been issued or until user ID FRANK logs on again, is autologged, or reconnects.
4. When you issue the RLIST command for a VMXEVENT profile, the output shows the members that have been added to the profile.

Activating an Individual VM Event Profile

An individual VM event profile will be activated or refreshed automatically whenever the user logs on, is autologged, or reconnects. You may also refresh the profile while the user is currently logged on by using the SETEVENT REFRESH command. For example, the following command resets the control and auditing options for user FRANK:

```
SETEVENT REFRESH USERSEL.FRANK
```

Note that the user FRANK must be logged on when the SETEVENT REFRESH is issued, or an error message will be displayed.

Suspending an Individual VM Event Profile

Use the SETEVENT RESET command to return a user to the auditing and authorization checking set in a system VM event profile. For example, to discontinue individual auditing and authorization checking for user ID FRANK, enter the following command:

```
SETEVENT RESET USERSEL.FRANK
```

Note: The SETEVENT RESET command does not delete the individual VM event profile for user ID FRANK. Issuing the command simply means that you *temporarily suspend* the use of the individual VM event profile that has been established for user ID FRANK. The suspension will stay in effect until you issue a REFRESH for user ID FRANK's individual VM event profile or until user ID FRANK next logs on, is autologged, or reconnects.

Deleting an Individual VM Event Profile

Use the RDELETE and SETEVENT RESET commands to not merely suspend, but to *delete* an individual VM event profile. For example, to delete user ID FRANK's individual VM event profile and have user ID FRANK be subject to the system VM event profile that is in effect on the system, follow this sequence. First, delete the user's individual VM event profile to ensure that the user's individual VM event profile will not be reactivated at LOGON. Enter the following command:

```
RDELETE VMXEVENT USERSEL.FRANK
```

Second, issue the SETEVENT RESET command to deactivate the user's individual VM event profile. The high-level qualifier is required when issuing this command, even though the previous command has in fact removed the individual VM event profile.

```
SETEVENT RESET USERSEL.FRANK
```

You would use this same sequence to delete the exempt status of a user by removing the user's individual VM event profile, thus making that user subject to the system VM event profile.

Note: You can specify RDELETE without specifying SETEVENT RESET and thus allow the user to be under the options of the user's individual VM event profile until the user logs off. However, if the user simply disconnects, and does not logoff, auditing and authorization checking for the user will continue to be done through the individual VM event profile. The safer course of action, if you want to remove the use of an individual VM event profile, is to issue the RDELETE and SETEVENT RESET commands in sequence.

Making a User Exempt

An installation may exempt a particular user from almost all command control or auditing; however, the installation must be aware of the implications. To exempt a user, the installation creates an individual VM event profile with a member list that specifies that all controllable VM events are not to be controlled (do not require authorization checking) and the audit options are not set for any VM events. For an accurate and up-to-date list of controllable VM events, issue the SETEVENT LIST command.

When the exempt user ID enters the system (by means of the LOGON, AUTOLOG or XAUTOLOG command), RACF will recognize that nothing is being audited or controlled for this user; at this point VM will no longer call RACF for authorization checking during the exempt user's course of operation. As a result of reduced RACF calls, performance may be improved. However, the user ID will still be subject to CP directory authorization, as if RACF were not installed.

When subsequent CP commands are processed for the exempt user ID, VM will not invoke RACF for any AUDIT, MAC, or DAC requests, except for a small subset of events for which RACF must be invoked (AUTOLOG, LOGON, LOGOFF, and XAUTOLOG).

Attention

Exempt profiles may be created for any user, at the installation's discretion. Keep in mind, however, that if an exempt profile is created for a service machine that does work on behalf of other users (as, for example, a batch machine), then jobs submitted to that machine will be executed with virtually no authorization checking or auditing being performed by RACF.

In a B1 configuration, the definition of such exempt individual VM event profiles must be limited to the set of trusted servers, and no one else. See *VM/ESA C2/B1 Trusted Facility Manual for VM/ESA with RACF* for a discussion of trusted servers.

For example, to define a VMXEVENT profile for the exempt user ID named SERVER1, use the RDEFINE command as follows:

```
RDEFINE VMXEVENT USERSEL.SERVER1
```

This will create a profile which by default contains no audited members and for which control is on for all controllable events. For the controllable events, members must be added explicitly to disable control for these events:

```
RALTER VMXEVENT USERSEL.SERVER1 ADDMEM(LINK/NOCTL)
RALTER VMXEVENT USERSEL.SERVER1 ADDMEM(COUPLE.G/NOCTL)
RALTER VMXEVENT USERSEL.SERVER1 ADDMEM(STORE.C/NOCTL)
RALTER VMXEVENT USERSEL.SERVER1 ADDMEM(TAG/NOCTL)
RALTER VMXEVENT USERSEL.SERVER1 ADDMEM(TRANSFER.D/NOCTL)
RALTER VMXEVENT USERSEL.SERVER1 ADDMEM(TRANSFER.G/NOCTL)
RALTER VMXEVENT USERSEL.SERVER1 ADDMEM(TRSOURCE/NOCTL)
RALTER VMXEVENT USERSEL.SERVER1 ADDMEM(DIAG0D4/NOCTL)
RALTER VMXEVENT USERSEL.SERVER1 ADDMEM(DIAG0E4/NOCTL)
RALTER VMXEVENT USERSEL.SERVER1 ADDMEM(APPCPWVL/NOCTL)
RALTER VMXEVENT USERSEL.SERVER1 ADDMEM(MDISK/NOCTL)
RALTER VMXEVENT USERSEL.SERVER1 ADDMEM(RSTDSEG/NOCTL)
RALTER VMXEVENT USERSEL.SERVER1 ADDMEM(DIAG0A0/NOCTL)
RALTER VMXEVENT USERSEL.SERVER1 ADDMEM(DIAG280/NOCTL)
```

Notes:

1. This list can change because of product updates. For an accurate and up-to-date list of controllable VM events, issue the SETEVENT LIST command.
2. A REXX exec can be created to define these profiles for several exempt user IDs.

The profile is put into effect the next time the user ID is logged on, is autologged, or reconnects. To activate the profile while the user ID is logged on, use the SETEVENT command. For example:

```
SETEVENT REFRESH USERSEL.SERVER1
```

Considerations for User IDs Autologged During System Initialization

Certain user IDs are logged on during system initialization. These user IDs include the following:

- AUTOLOG1— Logged on automatically at IPL time to perform functions you select
- OPERACCT— For CP *ACCOUNT system service to accumulate and checkpoint accounting records
- OPEREREP— For CP *LOGREC system service to accumulate and checkpoint error records
- OPERSYMP— For CP *SYMPTOM system service to accumulate and checkpoint symptom records
- OPERATOR— Primary system operator's user ID.

The autologged user IDs are tailorable using a system configuration file called SYSTEM CONFIG. RACF is not active on the system until after these user IDs have been logged on during system initialization. At this point, RACF has not been called to authenticate them; therefore, individual VM event profiles are not in effect for any of them, with the exception of the OPERATOR user ID. RACF does activate an individual VM event profile for the system operator, but not for the others. If your installation has an individual VM event profile for a user autologged during system IPL (besides OPERATOR), a SETEVENT REFRESH must be issued to activate the profile.

Because RACF autologs the user ID, AUTOLOG2, this SETEVENT command could be added to the PROFILE EXEC of AUTOLOG2. For example, if your installation has an individual VM event profile for the OPERACCT user, the following statement can be added to the PROFILE EXEC of AUTOLOG2:

```
RAC SETEVENT REFRESH USERSEL.OPERACCT
```

To authorize the use of the SETEVENT command for AUTOLOG2, you must ensure the AUTOLOG2 user ID has access to the RAC command processor and has the RACF SPECIAL and/or AUDITOR attribute, depending on what settings are being activated in the individual VM event profile.

Until RACF is active, all VM requests (VM events) from these autologged users are exempt from security authorization and auditing. Once RACF becomes active, VM requests from these users are handled using authorization and auditing rules which are in effect for that user or all users on the system.

For more information on SYSTEM CONFIG, see *z/VM Planning and Administration.*

Controllable VM Events

You can use the SETEVENT LIST command to generate a list of controllable and auditable VM events.

In the SETEVENT LIST output, examine the “CONTROLLABLE VM EVENTS” section of the output. (VM events listed in the “AUDITABLE VM EVENTS” section are not necessarily controllable.)

In the SETEVENT LIST output, “VM EVENT” indicates the name of the VM event as RACF recognizes it. “STATUS” indicates one of the following:

- NO_CONTROL indicates that VM is not to call RACF when the VM event occurs. This means that any user whose VM privilege class allows the use of the event in question can issue the command or code without checking a RACF resource profile.
- CONTROL indicates that VM is to call RACF when the VM event occurs. The call to RACF serves as an additional authorization check that occurs only if the user passes authorization checks performed by CP. For example, the user must have an appropriate VM privilege class and/or directory authorization before RACF is called.

Normal RACF authorization considerations, such as the use of security labels, can affect the granting of authority when these profiles are checked.

Note: If the command is prefixed with “RAC,” the output of the command is written to a file on your disk or directory accessed as A. You can print this file for study, and edit it to generate the appropriate RACF commands to protect selected VM events.

Commands issued before logon (such as DIAL, UNDIAL, and MESSAGE) have no user ID associated with them. To prevent this, see “Preventing Use of DIAL, UNDIAL, and MESSAGE Commands Before Logon” on page 163.

Sample output from the SETEVENT LIST command is shown in Figure 17 on page 162.

```
PRE-LOGON COMMANDS

COMMAND              CONFIGURED IN
-------              -------------
DIAL                         YES
MESSAGE.ANY                  YES
UNDIAL                       YES

CONTROLLABLE VM EVENTS

VM EVENT               STATUS     VM EVENT               STATUS
--------               ------     --------               ------
COUPLE.G               CONTROL    LINK                   NO_CONTROL
STORE.C                CONTROL    TAG                    CONTROL
TRANSFER.D             CONTROL    TRANSFER.G             CONTROL
TRSOURCE               NO_CONTROL DIAG0A0                CONTROL
DIAG0D4                CONTROL    DIAG0E4                CONTROL
DIAG280                CONTROL    APPCPWVL               CONTROL
MDISK                  CONTROL    RSTDSEG                CONTROL

AUDITABLE VM EVENTS

VM EVENT               STATUS     VM EVENT               STATUS
--------               ------     --------               ------
ACNT                   NO_AUDIT   ACTIVATE               NO_AUDIT
ADJUNCT                NO_AUDIT   ADSTOP                 NO_AUDIT
ASSOCIATE              NO_AUDIT   ATTACH                 NO_AUDIT
ATTN                   NO_AUDIT   AUTOLOG.A              NO_AUDIT
AUTOLOG.B              NO_AUDIT   BACKSPACE              NO_AUDIT
BEGIN                  NO_AUDIT   CACHE                  NO_AUDIT
CHANGE.D               NO_AUDIT   CHANGE.G               NO_AUDIT
CLOSE                  NO_AUDIT   COMMANDS               NO_AUDIT
COMMIT                 NO_AUDIT   CONCOPY                NO_AUDIT
COUPLE.G               AUDIT      CPACCESS               NO_AUDIT
CPCACHE                NO_AUDIT   CPHX                   NO_AUDIT
CPLISTFILE             NO_AUDIT   CPRELEASE              NO_AUDIT
CPFORMAT               NO_AUDIT   CPTRAP                 NO_AUDIT
 . . .
QUERY.VIRTUAL.G        NO_AUDIT   SET.ABEND              NO_AUDIT
SET.ACCOUNT            NO_AUDIT   SET.ACNT               NO_AUDIT
SET.AFFINITY           NO_AUDIT   SET.ASSIST             NO_AUDIT
 . . .
DIAG000                NO_AUDIT   DIAG004                NO_AUDIT
DIAG008                NO_AUDIT   DIAG00C                NO_AUDIT
DIAG010                NO_AUDIT   DIAG014                NO_AUDIT
 . . .
SPF_DELETE             NO_AUDIT   SPF_OPEN               NO_AUDIT
SDF_CREATE             NO_AUDIT   SDF_DELETE             NO_AUDIT
SDF_OPEN               NO_AUDIT   UTLPRINT               NO_AUDIT
MDISK                  NO_AUDIT   MAINTCCW               NO_AUDIT
RSTDSEG                NO_AUDIT
```

Figure 17. Sample Output from the SETEVENT LIST Command. This list can change because of product updates. For an accurate and up-to-date list, issue the SETEVENT LIST command.

Preventing Use of DIAL, UNDIAL, and MESSAGE Commands Before Logon

When a user issues a command before logging on, no user ID is associated with the command. This can make it difficult to determine who the user is. You can prevent users from issuing the DIAL, UNDIAL, and MESSAGE commands before logging on. To do this, issue the SETEVENT command with the NODIAL and NOPRELOGMSG operands, as follows:

```
SETEVENT  NODIAL  NOPRELOGMSG
```

Protecting VM Resources

The following sections describe how to create profiles to protect resources on VM systems. Table 17 refers you to the specific sections in this chapter for each type of resource:

Table 17. Where to Find Specific Resource Information

Types of Resources	Chapter Reference
General information	“Defining Profiles for General Resources”
Minidisks	“Protecting VM Minidisks” on page 165
Virtual unit record devices	“Protecting Virtual Unit Record Devices” on page 169
RSCS nodes	“Protecting RSCS Nodes” on page 171
Alternate user IDs	“Protecting Alternate User IDs” on page 172
Restricted segments	“Protecting Restricted Segments with the VMSEGMT Class” on page 174
Guest LANs	“Protecting Guest LANs and Virtual Switches” on page 176
Terminals	“Protecting Terminals on VM” on page 180
Certain CP commands, diagnose codes, and other functions	“Protecting the Use of CP Commands, Diagnose Codes, and Other Functions” on page 184

Defining Profiles for General Resources

To protect a general resource, use the RDEFINE command to define a general resource profile. You can also use the ISPF panels to define general resource profiles.

When you create a general resource profile, you must specify the class name and the profile name. For example:

```
RDEFINE  class-name  profile-name
```

Any time you wish to refer to the profile (for example, when changing its access list), you must give the profile name and class name.

Examples in this book also include the UACC (universal access authority):

```
RDEFINE  class-name  profile-name  UACC(universal-access-authority)
```

UACC is usually shown as NONE. This prevents all users not otherwise specified in the access list from accessing the resource.

Usually, you will also issue the PERMIT command to set up the access list in the profile. A sample PERMIT command is:

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

When you enter the RDEFINE command, you can specify much more than just profile name, class name, and UACC. In most cases, RACF provides appropriate defaults for this additional information. Where additional information is necessary for the profile (such as specifying the ADDMEM operand for resource grouping profiles), this book gives examples and describes appropriate values. Some of the additional operands that you might consider specifying are:

- OWNER—The user ID or group name of the owner of the profile
- NOTIFY—A user ID to be notified when access attempts fail
- AUDIT—Whether access attempts are to be logged, and if so, at which level.

Other operands are available. For a complete list of the operands for the RDEFINE command, see *RACF Command Language Reference*.

Protecting General Resources with Discrete Profiles

A *discrete profile* protects a single resource. For example, if a resource requires special access authorization or unique logging information, you can protect it with a discrete profile.

You can protect a resource with a discrete profile by using the RDEFINE command with the resource's class name and profile name. For a description of the IBM-supplied general resource classes, see "Protecting General Resources" on page 20.

The following example shows how to protect SMITH's A-disk (which has a virtual address of 191) with a discrete profile:

```
RDEFINE  VMMDISK  SMITH.191  UACC(NONE)
```

Protecting General Resources with Generic Profiles

A *generic profile* protects one or more resources. Resources protected by generic profiles must have similar names and identical security requirements. For example, a generic minidisk profile can protect one or more minidisks.

If you use a generic profile on VM, follow the naming conventions for VM resources and the rules for defining generic profiles. To protect a resource with a generic profile, use the RDEFINE command and specify the resource's class name and generic profile name. The *generic profile name* must contain one or more generic characters (%, *, or &). For example, you can protect all VM minidisks owned by USERA with a generic profile as follows:

```
RDEFINE  VMMDISK  USERA.*  UACC(NONE)
```

Note: A SETROPTS GENERIC(classname) command must first be issued to turn generics on for the class, followed by a SETROPTS REFRESH command.

Protecting General Resources Using Models from Existing Profiles (or Access Lists in Existing Profiles)

You can protect a resource with a generic or discrete profile by using an existing generic or discrete profile as a model. Create the model by specifying the existing profile name on the FROM operand.

- As you create a new profile, you can specify FROM on the RDEFINE command to identify the profile to be copied. For example, to copy an existing generic minidisk profile called SMITH.* when creating a new minidisk profile named JONES.*:

```
RDEFINE  VMMDISK  JONES.*  FROM(SMITH.*) UACC(NONE)
```

If the profile you are copying is not in the same class, specify the FCLASS operand. The following command copies information from profile TV1234 in class TAPEVOL:

```
RDEFINE  VMMDISK  JONES.*  FROM(TV1234)  FCLASS(TAPEVOL) UACC(NONE)
```

- When you work with an access list, you can specify FROM on the PERMIT command to identify a profile whose access list is to be copied. For example, to copy the access list of an existing minidisk profile called SMITH.* to an existing minidisk profile named JONES.*:

  ```
  PERMIT  JONES.*  CLASS(VMMDISK)
          FROM(SMITH.*)
  ```

  If the profile you are copying is not in the same class, specify the FCLASS operand. The following command copies the access list from profile TV1234 in class TAPEVOL:

  ```
  PERMIT  JONES.*  CLASS(VMMDISK)
          FROM(TV1234)  FCLASS(TAPEVOL)
  ```

Note: The copied information is not necessarily identical to that specified in the FROM profile. See “Possible Changes to Copied Profiles When Modeling Occurs” on page 46.

Note to the Reader

General information on generic profiles for general resources on VM is in “Generic Profile Names” on page 93.

Information on global access checking is in Chapter 6, “Setting Up the Global Access Checking (GAC) Table” on page 103.

Access Authorization Checking for the General Resources

During access-authorization checking, RACF first checks for a discrete profile for a resource. If a discrete profile does not exist, RACF examines the generic profiles in the order of *most specific to least specific* profile name. Therefore, if a discrete profile does not exist, RACF uses the most specific matching generic profile. For a description of the search order RACF uses, see Table 12 on page 96.

Protecting VM Minidisks

You can use RACF to control who can link to VM minidisks using profiles in the VMMDISK resource class. VM calls RACF for an authorization check in the VMMDISK class for two different events:

1. LINK command—A user's attempt to link to another user's minidisk.
2. MDISK event—A user linking to his or her own minidisk. MDISK events occur at logon time when the user's MDISK directory statements are being processed, or when the user issues a LINK command for a self-owned minidisk.

Public Minidisks: Your installation may have many minidisks that do not contain installation-sensitive data and that are frequently accessed by large numbers of users in READ mode. You can consider using the global access checking table or creating a global minidisk table to help performance. These tables are applicable to public minidisks your installation has identified.

For information on the global minidisk table, see *RACF Macros and Interfaces* and *SPL: RACF*.

For information on the global access checking table, see Chapter 6, “Setting Up the Global Access Checking (GAC) Table” on page 103.

Profile Considerations in the VMMDISK Class

Decide on the type of minidisk profiles you want to create. There are several things to consider.

Generic Profiles: You can use generic profiles for VM minidisks. If you do so, it is strongly recommended that you issue the following command:

```
SETROPTS GENLIST(VMMDISK)
```

This command causes one copy of each generic profile for the VMMDISK class to be kept in the RACF service machine.

Changes made to a generic minidisk profile may not take effect until the in-storage copy kept in the RACF service machine is refreshed using the SETROPTS GENERIC(VMMDISK) REFRESH command.

Because general users cannot issue the SETROPTS command, using generic minidisk profiles can prevent users from promptly and effectively changing minidisk profiles. For more information on when changes to minidisk profiles take effect, see “When Changes to Minidisk Profiles Take Effect” on page 306.

Discrete Profiles: You can use discrete profiles for VM minidisks. It is recommended that you use discrete profiles for those minidisks for which you anticipate more frequent profile updates.

Changes made to a discrete minidisk profile immediately affect attempts to LINK to the protected minidisk.

ACIGROUP Considerations: If a user's directory entry has an ACIGROUP control statement, the profile names for the user's minidisks are prefaced with the user's ACIGROUP name. See “Use of ACIGROUP Control Statements” on page 194.

Procedure to Protect Minidisks

To protect VM minidisks, do the following:

1. Create profiles to protect minidisks:

   ```
   RDEFINE VMMDISK userid.virtual-address
   ```

   where:

   userid is the owning user ID.

   virtual-address is the virtual address of the minidisk as defined in the user's CP directory entry

You can define a minidisk with a 4-character virtual address; however, RACF does not allow the first character to be 0. For example, RACF allows SMITH.191, SMITH.1234, and SMITH.002, but does not allow SMITH.0191. To create a minidisk profile for SMITH's A-disk, which would have a virtual address of 191, use the following command:

```
RDEFINE VMMDISK   SMITH.191  UACC(NONE)
```

2. Use the PERMIT command to grant or deny users and groups access to minidisks. Ensure that the owner of the minidisk is included in the access list so that, when the users link to their own minidisks—either with the LINK command or while logging on—the authorization will not fail.

   ```
   PERMIT SMITH.191 CLASS(VMMDISK) ID(SMITH) ACCESS(ALTER)
   ```

   For advice on what authorization to request in the PERMIT command, see "RACF Minidisk Access Authorities."

3. Activate the VMMDISK class:

   ```
   SETROPTS CLASSACT(VMMDISK)
   ```

4. Make sure protection of LINK and MDISK is active. It is active by default if an installation has not changed the setting in the currently active VMXEVENT profile by issuing, for example:

   ```
   RDEFINE VMXEVENT EVENTS1 ADDMEM(LINK/NOCTL MDISK/NOCTL)
   ```

   If protection of LINK and MDISK is not currently active, activate it by issuing:

   ```
   RALTER VMXEVENT EVENTS1 DELMEM(LINK/NOCTL MDISK/NOCTL)
   SETEVENT REFRESH EVENTS1
   ```

Attention

You should be aware that, because self-LINKs are frequently used, controlling MDISK can have a significant effect on system performance.

RACF Minidisk Access Authorities

Access authorities correspond to access modes that are specified on the CP LINK command. The following list shows RACF access authorities and their corresponding CP LINK access modes:

NONE Does not allow users to access the minidisk.

ATTENTION

Anyone who has READ, UPDATE, CONTROL, or ALTER authority to a protected minidisk can create copies of the data files on it. If users copy the data files to a minidisk for which they can control the security characteristics, they can downgrade the security characteristics of the copied files. For this reason, you will probably want to assign a UACC of NONE, and then selectively permit a small number of users to access your minidisk, as their needs become known. (See *RACF General User's Guide* for information on how to permit selected users or groups to access a minidisk.)

READ Allows users to access the minidisk for reading or copying only. This enables users to request an access mode of read (R), read-read (RR), stable-read (SR), or exclusive-read (ER) on the CP LINK command. (Note that users who can read files on a minidisk can copy or print them.)

UPDATE Allows users to read from, copy from, or write to the minidisk. This enables users to request an access mode of write (W), write-read (WR), stable-write (SW), or exclusive-write (EW) on the CP LINK command.

CONTROL Allows users to read from, copy from, or write to the minidisk. This enables users to request an access mode of multiple (M), multiple-read (MR), or stable-multiple (SM) on the CP LINK command.

ALTER Allows users to read from, copy from, or write to the minidisk. This enables users to request an access mode of multiwrite (MW) on the CP LINK command.

When specified in a discrete profile, ALTER allows users to read, alter, and delete the profile itself *including the access list*. However, ALTER does not allow users to change the owner of the profile.

When specified in a generic profile, ALTER gives users *no* authority over the profile itself.

Note: For a description of the different CP LINK access modes, refer to *z/VM CP Command and Utility Reference*.

SYSSEC Considerations for LINK and MDISK

The SYSSEC macro, coded in the RACF module HCPRWA, can affect the final outcome of resource requests in the VMMDISK class. The SYSSEC macro defines the relationship between RACF's response to a resource access request and the final disposition of that request by VM. This relationship changes as you change the SYSSEC statement in HCPRWA.

For example, a user may link to a minidisk which is not protected by a RACF profile. SYSSEC can be coded so that for this case, RACF will do one of the following:

- Allow access
- Disallow access
- Defer the access decision to VM

You should know how the SYSSEC macro is coded on your system. See *RACF Macros and Interfaces* for more information.

Security Label Checking for LINK and MDISK Events

When the SECLABEL class is active, VM calls RACF for authorization checking for a link to a minidisk even if control is turned off for the LINK and MDISK events.

If the SECLABEL class is active, but you do not want to use RACF minidisk protection, deactivate the VMMDISK class by issuing:

```
SETROPTS NOCLASSACT(VMMDISK)
```

VM still calls RACF for MAC authorization, but RACF defers the authorization to VM. VM will then perform LINK authorization based on the CP directory.

The type of SECLABEL authorization required is based on the link mode the user requested for the minidisk.

1. If the user is requesting a link mode that requires READ access to the VMMDISK profile, a read-only authorization request is performed.
2. If the user is requesting a link mode that requires UPDATE, CONTROL, or ALTER access to the VMMDISK profile, a read/write authorization request is performed.

Note: When the SECLABEL class is active and SETROPTS MLS(FAILURES) is in effect, RACF *downgrades* MDISK requests for MR or WR access if the SECLABEL of the user requesting the link does not equal the SECLABEL of the minidisk. In this case, the user receives read-only access to the minidisk, but only if so authorized.

The outcome of the request depends on the SETROPTS options that are in effect. For more information, see “Security Label Authorization Checking” on page 315.

Additionally, when the SECLABEL class is active and SETROPTS MLACTIVE(FAILURES) has been issued, profiles in the VMMDISK class are required to contain a SECLABEL. Therefore, authorization requests for a minidisk which does not have an assigned SECLABEL will fail.

Protecting Virtual Unit Record Devices

You can use RACF to control which users can send files to virtual unit record devices using profiles in the VMRDR resource class. Virtual unit record devices are the readers, punches, and printers for any virtual machine.

When protecting virtual unit record devices, RACF is called for authorization for several events. When control is turned on for the TRANSFER.G command, the other commands that are protected include:

- TRANSFER and CHANGE TO (for spool files you own or originated)
- CLOSE TO
- SPOOL TO
- SPOOL FOR
- VMDUMP TO
- DIAG094 (DIAGNOSE code X'94' with the TO parameter)

The commands protected by RACF when control is turned on for TRANSFER.D are:

- TRANSFER and CHANGE TO (for spool files you do not own and did not originate)
- TRSAVE TO

You can use this process to control who can send files for processing by batch machines or networking machines.

Notes:

1. RACF protects all unit record devices for a particular virtual machine in the same manner. For example, if USERA can send files to USERB's reader, USERA can also send files to USERB's punch and printer. Conversely, if USERA cannot send files to USERB's reader, USERA also cannot send files to USERB's punch or printer.
2. While installing RACF, your installation can create a VMRDR profile for each user whose virtual reader is defined in the VM directory. As you add new users to the system, you might need to manually create VMRDR profiles for them. (For more information on defining user IDs during installation, see *RACF Program Directory*.)

To define and protect a user's virtual unit record devices with RACF, take the following steps:

1. Create a profile in the VMRDR class:

   ```
   RDEFINE  VMRDR  userid  UACC(NONE)
   ```

   where *userid* is the owning user ID.

   Note: If a user's directory entry has an ACIGROUP control statement, the profile names for the user's virtual unit record devices must be prefaced with the user's ACIGROUP name. See “Use of ACIGROUP Control Statements” on page 194.

2. To allow users to send files to readers, printers, and punches protected by the profile, give them UPDATE access to the profile:

   ```
   PERMIT  userid  CLASS(VMRDR)  ID(user or group)  ACCESS(UPDATE)
   ```

3. Activate the VMRDR class:

   ```
   SETROPTS CLASSACT(VMRDR)
   ```

4. Make sure protection of TRANSFER.D and TRANSFER.G is active. It is active by default if an installation has not changed the setting in the currently active VMXEVENT profile by issuing, for example:

   ```
   RDEFINE VMXEVENT EVENTS1 ADDMEM(TRANSFER.D/NOCTL TRANSFER.G/NOCTL)
   ```

   If protection of TRANSFER.D and TRANSFER.G is not currently active, activate it by issuing:

   ```
   RALTER VMXEVENT EVENTS1 DELMEM(TRANSFER.D/NOCTL TRANSFER.G/NOCTL)
   SETEVENT REFRESH EVENTS1
   ```

Attention

You should be aware that, because these commands are frequently used, controlling their use can have a significant effect on system performance.

SYSSEC Considerations for Unit Record Devices

The SYSSEC macro, coded in the RACF module HCPRWA, can affect the final outcome of resource requests in the VMRDR class. The SYSSEC macro defines the relationship between RACF's response to a resource access request and the final disposition of that request by VM. This relationship changes as you change the SYSSEC statement in HCPRWA.

For example, a user may send a file to a virtual unit record device which is not protected by a RACF profile. SYSSEC can be coded so that for this case, RACF will do one of the following:

- Allow access
- Disallow access
- Defer the access decision to VM

You should know how SYSSEC is affecting your VMRDR requests. See *RACF Macros and Interfaces* for more information.

Security Label Considerations for Unit Record Devices

If a security label exists in the VMRDR profile protecting the resource, and the SECLABEL class is active, RACF compares the security label of the user with the security label contained in the profile. RACF grants access if the security label of the user is equal to or higher than the security label contained in the profile.

Protecting RSCS Nodes

You can use RACF to control which users can send files to specific RSCS nodes using the TAG DEVICE or TAG FILE commands.

To control access to nodes, do the following:

1. Use the RDEFINE command to define a profile for the node:

   ```
   RDEFINE  VMNODE  node-id  UACC(NONE)
   ```

   where *node-id* is the node id defined in the RSCS service machine.

2. Use the PERMIT command to give appropriate users and groups UPDATE access authority to the profile:

   ```
   PERMIT  node-id  CLASS(VMNODE)  ID(user or group)  ACCESS(UPDATE)
   ```

   UPDATE access authority is required for a user to send files to the node.

3. Activate the VMNODE class:

   ```
   SETROPTS CLASSACT(VMNODE)
   ```

4. Make sure protection of TAG is active. It is active by default if an installation has not changed the setting in the currently active VMXEVENT profile by issuing, for example:

   ```
   RDEFINE VMXEVENT EVENTS1 ADDMEM(TAG/NOCTL)
   ```

   If protection of TAG is not currently active, activate it by issuing:

   ```
   RALTER VMXEVENT EVENTS1 DELMEM(TAG/NOCTL)
   SETEVENT REFRESH EVENTS1
   ```

Attention

You should be aware that, because the TAG command is frequently used, controlling its use can have a significant effect on system performance.

SYSSEC Considerations for RSCS Nodes

The SYSSEC macro, coded in the RACF module HCPRWA, can affect the final outcome of resource requests in the VMNODE class. The SYSSEC macro defines the relationship between RACF's response to a resource access request and the final disposition of that request by VM. This relationship changes as you change the SYSSEC statement in HCPRWA.

For example, a user may send a file to an RSCS node which is not protected by a RACF profile. SYSSEC can be coded so that for this case, RACF will do one of the following:

- Allow access
- Disallow access
- Defer the access decision to VM

You should know how SYSSEC is affecting your VMNODE requests. See *RACF Macros and Interfaces* for more information.

Security Label Considerations for RSCS Nodes

If a security label exists in the VMNODE profile protecting the resource, and the SECLABEL class is active, RACF compares the security label of the user with the security label contained in the profile, and grants access if the security label of the user is equal to or higher than the security label contained in the profile.

Additionally, when the SECLABEL class is active, an authorization check is made in the VMMAC class for the TAG FILE command. See “Protecting VM Events with the VMMAC Class” on page 192 for more information.

Protecting Alternate User IDs

The alternate user ID function is a VM facility that enables one virtual machine to act with the access authority of another virtual machine. This relationship is established by use of the VM Diagnose code X'D4'. In most cases, this diagnose code is issued by a batch virtual machine (hereafter referred to as the “master”). This allows a batch processor (or “worker”) to perform work for an end user (or “alternate user”).

You can use RACF to protect which workers will be able to perform work on behalf of a particular alternate user through the use of profiles in the VMBATCH resource class.

The ICHRCX02 RACHECK postprocessing exit is shipped with the RACF product. This exit allows an alternate user to access data that the worker can access, but the alternate user cannot normally access. It can be used, for example, to allow users to access a restricted compiler only when submitting a batch job to a specified batch machine. If this RACHECK postprocessing exit is not desirable, remove the ICHRCX02 exit (see *SPL: RACF* for more information on this procedure).

To allow the use of alternate user IDs, take the following steps:

1. To allow a new user to use the alternate user ID function, create a profile for that user in the VMBATCH class as follows:

```
RDEFINE  VMBATCH  userid  UACC(NONE)
```

Note: While installing RACF, your installation can create a VMBATCH profile for each user defined in the VM directory. As you add new users to the system, you define their user IDs to RACF in the VMBATCH resource class if those users intend to use the alternate user ID function. (For more information on defining user IDs during installation, see *RACF Program Directory for VM Installations*.)

2. Authorize that user ID to be an alternate user ID for a worker machine in one of the following ways:

   - Use the PERMIT command to give the worker CONTROL access authority to the VMBATCH profile for the alternate user ID:

     ```
     PERMIT  alt-userid  CLASS(VMBATCH)  ID(worker)  ACCESS(CONTROL)
     ```

     For example, to allow worker batch machine BAT1 to perform work on behalf of user USER1, issue the following command:

     ```
     PERMIT  USER1  CLASS(VMBATCH)  ID(BAT1)  ACCESS(CONTROL)
     ```

     This allows USER1 to submit batch jobs either to worker batch machine BAT1 directly, or to the master batch machine, who in turn would assign the job to worker machine BAT1.

   - You can also organize worker batch machines into groups and authorize them all to be able to perform work on behalf of an alternate user. Take the following steps:

     a. Define a group for the worker machines. For example, the following command defines group BATGROUP:

        ```
        ADDGROUP  BATGROUP
        ```

     b. Connect each worker machine to the group. For example, the following command connects the worker machines BAT1, BAT2, and BAT3 to the group BATGROUP:

        ```
        CONNECT  (BAT1  BAT2  BAT3)  GROUP(BATGROUP)
        ```

     c. Allow these workers to perform work on behalf of the alternate user:

        ```
        PERMIT alt-userid CLASS(VMBATCH) ID(BATGROUP) ACCESS(CONTROL)
        ```

     d. To ensure access will be granted with this method, you must have either list-of-groups access checking activated:

        ```
        SETROPTS GRPLIST
        ```

        or change the alternate user's default group to match the group name you specified in the previous PERMIT example:

        ```
        ALTUSER alt-userid DFLTGRP(BATGROUP)
        ```

3. Activate the VMBATCH class:

   ```
   SETROPTS CLASSACT(VMBATCH)
   ```

4. Make sure protection of DIAG0D4 is active. It is active by default if an installation has not changed the setting in the currently active VMXEVENT profile by issuing, for example:

   ```
   RDEFINE VMXEVENT EVENTS1 ADDMEM(DIAG0D4/NOCTL)
   ```

   If protection of DIAG0D4 is not currently active, activate it by issuing:

   ```
   RALTER VMXEVENT EVENTS1 DELMEM(DIAG0D4/NOCTL)
   SETEVENT REFRESH EVENTS1
   ```

Security Label Considerations for Alternate User IDs

When the SECLABEL class is active (even if control has been turned off for the DIAG0D4 event using a VMXEVENT profile and the SETEVENT command), RACF verifies that both the worker machine and the alternate user ID have access to the specified SECLABEL, which appears in the parameter list for DIAGNOSE X'D4', subcode X'04'. However, these RACF checks do not ensure that the worker machine is running at a SECLABEL equivalent to the alternate user ID's SECLABEL.

To enforce SECLABEL checking, an installation should have a worker batch machine available for each SECLABEL used by alternate user IDs, and the worker batch machine should only have authorization to that one SECLABEL. The SECLABEL of the work done by each worker machine should be equivalent to the SECLABEL at which the alternate user is logged on.

When the SECLABEL class is active, VM does not allow DIAG0D4 subcode X'00'.

RACF compares the SECLABEL of the worker machine with the SECLABEL contained in the VMBATCH profile when all of the following conditions are true:

- The SECLABEL class is active
- Control is on for the DIAG0D4 event (it is on by default unless a VMXEVENT profile is active containing a DIAG0D4/NOCTL member)
- A SECLABEL exists in the VMBATCH profile for the alternate user ID.

RACF grants access if the SECLABEL of the worker machine is equal to or higher than the SECLABEL contained in the VMBATCH profile.

Allowing Batch Machines to Access a User's Minidisks

When a worker machine is running a user's job, and the job attempts to link to the user's minidisk, VM calls RACF for a LINK authorization check in the VMMDISK class. Note that this is a LINK request and not a self-link (or MDISK) since it is the worker machine issuing the LINK. Since the worker machine has the user's access authorities, the user must have the correct access authority to the minidisk (usually ALTER), or the LINK will fail.

For example, user SMITH can do the following:

```
PERMIT  SMITH.191  CLASS(VMMDISK)  ID(SMITH)  ACCESS(ALTER)
```

As security administrator, you can create an entry in the global access checking table that does the equivalent of the preceding PERMIT for all users on the system:

```
RALTER  GLOBAL  VMMDISK  ADDMEM(&RACUID.*/ALTER)
```

For more information on global access checking, see Chapter 6, “Setting Up the Global Access Checking (GAC) Table” on page 103.

Protecting Restricted Segments with the VMSEGMT Class

RACF can protect named saved segments (NSS) or discontiguous saved segments (DCSS) that are defined with the RSTD option of the CP DEFSEG or CP DEFSYS commands. To determine if your installation has any NSS's or DCSS's defined as restricted, issue QUERY NSS from a privilege class E user ID. Those segments for which an R is displayed under the CL column are defined as restricted. If you need to define restricted segments, refer to the DEFSEG or DEFSYS commands in *z/VM CP Command and Utility Reference*.

The segments are protected by defining RACF profiles in the VMSEGMT class. VM calls RACF for authorization checking on a restricted segment when one of the following events occurs:

- A user IPLs a restricted segment.
- DIAGNOSE code X'64' is issued to find or load a restricted segment. This includes subcodes X'0000', X'0004', X'000C', X'0010', and X'0018'.
- A user attempts to start a trace file (TRSOURCE ENABLE) for a VMGROUP. (This call is made for **all** named saved segments and **only** when the SECLABEL class is active.)

To use this protection for restricted segments with the VMSEGMT class, follow these steps:

1. Define RACF profiles for the segments to be protected.

   - For an NSS, the profile naming convention is *NSS.spacename*
   - For a DCSS, the profile naming convention is *DCSS.spacename*

   where *spacename* is the name of the space in which the segment resides and is specified when the segment is defined on the CP DEFSEG or DEFSYS command. A space can contain many DCSSs, but the entire space is loaded. In the case of an NSS, there is only one NSS per space.

   ```
   RDEFINE  VMSEGMT  profile_name UACC(NONE)
   ```

2. Authorize the appropriate users to use the restricted segment.

   ```
   PERMIT profile_name CLASS(VMSEGMT)
          ID(user or group) ACCESS(access_authority)
   ```

   UPDATE authority is required if a user must have shared write access to a DCSS or NSS; otherwise, READ authority is sufficient. In other words, if a DCSS or NSS is defined with at least one segment that has the SW or SN attribute, then UPDATE access is required. To start a trace file (TRSOURCE ENABLE) for a VMGROUP, a user must have UPDATE access to the VMSEGMT profile which protects the NSS associated with the VMGROUP.

3. Activate the VMSEGMT class.

   ```
   SETROPTS CLASSACT(VMSEGMT)
   ```

4. Make sure protection of RSTDSEG is active. It is active by default if an installation has not changed the setting in the currently active VMXEVENT profile by issuing, for example:

   ```
   RDEFINE VMXEVENT EVENTS1 ADDMEM(RSTDSEG/NOCTL)
   ```

   If protection of RSTDSEG is not currently active, activate it by issuing:

   ```
   RALTER VMXEVENT EVENTS1 DELMEM(RSTDSEG/NOCTL)
   SETEVENT REFRESH EVENTS1
   ```

Security Label Considerations in the VMSEGMT Class

When the SECLABEL class is active, VM calls RACF for authorization checking for restricted segments even if control is turned off for the RSTDSEG event.

If your installation does not want to use SECLABEL checking for the VMSEGMT class, deactivate the class by issuing:

```
SETROPTS NOCLASSACT(VMSEGMT)
```

VM still calls RACF for MAC authorization, but RACF defers the authorization to VM. VM treats this as a successful MAC authorization for the RSTDSEG event or TRSOURCE ENABLE event.

If the SECLABEL class is active and the SETROPTS MLACTIVE(FAIL) option is in effect, you must assign security labels to each of your VMSEGMT profiles by issuing:

```
RALTER VMSEGMT profile-name SECLABEL(seclabel-name)
```

Because segments are protected based on the space name, you should ensure all segment data within a space is at the same SECLABEL.

The type of SECLABEL authorization required is based on the RACF access mode required for accessing the restricted segment:

- For segments that require READ access to the VMSEGMT profile, a read-only authorization request is performed.
- For segments that require UPDATE access to the VMSEGMT profile, a read/write authorization request is performed.

The outcome of the request depends on the SETROPTS options that are in effect. For more information, see “Security Label Authorization Checking” on page 315.

Additionally, when the SECLABEL class is active and SETROPTS MLACTIVE(FAILURES) has been issued, profiles in the VMSEGMT class are required to contain a SECLABEL. Therefore, authorization requests for a segment which does not have an assigned SECLABEL will fail.

Protecting Guest LANs and Virtual Switches

RACF can be used to protect Guest LANs and virtual switches using profiles in the VMLAN class. From an access control perspective, guest LANs and virtual switches are treated the same way. In this document, the term "LAN" will be used to mean both guest LANs and virtual switches. When differences exist, they will be identified. For more information on Guest LANs and virtual switches, see *z/VM V5R1.0 Connectivity*.

Note:

VMLAN support is only available on z/VM Version 5 Release 1.0 and higher.

The VMLAN class contains two sets of profiles to protect LANs; base profiles that control the ability of a VM user to use a LAN, and, for IEEE VLAN-aware virtual switches, VLAN ID-qualified profiles that are used to assign a user to one or more IEEE VLANs.

Base Profiles

Base profiles are named *userid.name* where *userid* is the LAN owner and *name* is the name of the LAN. Both qualifiers will be eight characters or less. For example, guest LAN NET130 owned by TOMBRADY is represented by profile TOMBRADY.NET130. In the case of a virtual switch, *userid* will always be "SYSTEM". These profiles will control authorization and auditing of attempts by any user to COUPLE to a guest LAN or virtual switch. A user must have UPDATE access to the profile in order for the COUPLE to be authorized.

Note:

VMLAN support is only available on z/VM Version 5 Release 2.0 and higher.

Promiscuous Mode Authorization Promiscuous Mode is a mode of operation where the network adapter intercepts all data flowing over the network regardless of destination MAC or IP address. It is sometimes referred to as "LAN sniffing". This mode allows troubleshooting of potential networking problems using existing network debug tools such as **tcpdump** or **ethereal**. Since Promiscuous Mode in a guest virtual machine environment provides the guest NIC with a copy of all data traversing the LAN segment, authorization to use Promiscuous Mode should be given only when necessary. A user must be given CONTROL access to the base profile in order for Promiscuous Mode to be authorized. See *z/VM V5R2.0 Connectivity* for more information about using Promiscuous Mode to trouble shoot virtual networking problems.

VLAN ID-qualified profiles

If a virtual switch is VLAN aware (defined with the "VLAN *defvid*" parameter), then a secondary set of VLAN ID-qualified VMLAN profiles are used to control the ability of a virtual machine to become a member of a particular IEEE VLAN. These profiles are named SYSTEM.*name*.*vid*, where *name* is the name of the virtual switch and *vid* is a VLAN ID having a value between 1 and 4094, inclusive. The *vid* qualifier must consist of four decimal digits; leading zeroes must be entered for VLAN IDs less than four digits. See below for an example.

A user must have UPDATE access to a qualified profile to be considered authorized. No auditing will be performed as a result of checking the VLAN ID-qualified profiles and no violations will be reported. These profiles are consulted by RACF in order to build a list of authorized VLAN IDs on a given virtual switch for a given user. The list is returned to z/VM. The VLAN ID-qualified profiles are only consulted if the user has UPDATE access to the base profile protecting the virtual switch.

Note that z/VM allows the specification of a default VLAN ID when defining a virtual switch to z/VM. If the default is not specified, the value 1 is used. RACF does not have a mechanism for designating a default VLAN ID using profiles in the VMLAN class. If a base profile grants a user access to a virtual switch, but the user is not permitted to any VLAN ID-qualified profiles for that virtual switch, RACF will direct z/VM to authorize the user only to the default VLAN ID defined to z/VM for that virtual switch.

It is up to the administrator to ensure consistency between a base profile and its set of VLAN ID-qualified profiles. For example, assume a user has been granted UPDATE access to the base profile and to a set of VLAN ID-qualified profiles. If the user's access is removed from the base profile, RACF will not automatically remove the user's access from the VLAN ID-qualified profiles. The VLAN ID-qualified profiles will be ignored for this user.

Notes:

1. VLAN ID-qualified profiles must be discrete; generic profiles will be ignored.
2. Global Access Checking cannot be used for VLAN ID-qualified profiles.

Examples

Example 1

Define the VMLAN profile protecting the guest LAN named NET130, owned by TOMBRADY. Allow anyone in the PATRIOTS group to COUPLE to the LAN.

```
RAC RDEFINE VMLAN TOMBRADY.NET130 UACC(NONE)

RAC PERMIT TOMBRADY.NET130 CLASS(VMLAN) ID(PATRIOTS) ACCESS(UPDATE)
```

Note: After issuing the RDEFINEs and PERMITs as shown in Example 1 and Example 2, you must activate the VMLAN class:

```
SETROPTS CLASSACT(VMLAN)
```

Then make sure that protection of COUPLE.G is active. It is active by default if an installation has not changed the setting in the currently active VMXEVENT profile by issuing, for example:

```
RALTER VMXEVENT EVENTS1 ADDMEM(COUPLE.G/NOCTL)
```

If protection of COUPLE.G is not currently active, activate it by issuing:

```
RALTER VMXEVENT EVENTS1 DELMEM(COUPLE.G/NOCTL)
SETEVENT REFRESH EVENTS1
```

Example 2

In this example, an administrator wants to define the RACF profiles to represent the virtual switch named SWITCH01. Additionally, the administrator wants to assign user SUSAN to VLAN IDs 5, 10, and 15, WILL to VLAN ID 10, and MARYELEN to VLAN ID 1.

```
RAC RDEFINE VMLAN SYSTEM.SWITCH01 UACC(NONE)

RAC PERMIT SYSTEM.SWITCH01 CLASS(VMLAN) ID(SUSAN WILL MARYELEN)
           ACCESS(UPDATE)

RAC RDEFINE VMLAN SYSTEM.SWITCH01.0001 UACC(NONE)

RAC PERMIT SYSTEM.SWITCH01.0001 CLASS(VMLAN) ID(MARYELEN) ACCESS(UPDATE)

RAC RDEFINE VMLAN SYSTEM.SWITCH01.0005 UACC(NONE)

RAC PERMIT SYSTEM.SWITCH01.0005 CLASS(VMLAN) ID(SUSAN) ACCESS(UPDATE)

RAC RDEFINE VMLAN SYSTEM.SWITCH01.0010 UACC(NONE)

RAC PERMIT SYSTEM.SWITCH01.0010 CLASS(VMLAN) ID(SUSAN WILL)
           ACCESS(UPDATE)

RAC RDEFINE VMLAN SYSTEM.SWITCH01.0015 UACC(NONE)

RAC PERMIT SYSTEM.SWITCH01.0015 CLASS(VMLAN) ID(SUSAN) ACCESS(UPDATE)
```

See the note in Example 1 about activating the VMLAN class.

SYSSEC Considerations for Guest LANs

The SYSSEC macro, coded in the RACF module HCPRWA, can affect the final outcome of resource requests in the VMLAN class. The SYSSEC macro defines the relationship between RACF's response to a resource access request and the final disposition of that request by z/VM. This relationship changes as you change the SYSSEC statement in HCPRWA.

For example, a user may link to a Guest LAN which is not protected by a RACF profile. SYSSEC can be coded so that for this case, RACF will do one of the following:

- Allow access
- Disallow access
- Defer the access decision to z/VM

You should know how the SYSSEC macro is coded on your system. See *RACF Macros and Interfaces* for more information.

Security Label considerations for Guest LANs

When the SECLABEL class is active, z/VM calls RACF for authorization checking for Guest LANs, even if control is turned off for the COUPLE.G event. The SECLABEL check is performed in concert with the access check, and they cannot be controlled separately when the SECLABEL class is active. RACF performs a read/write SECLABEL check between the user's SECLABEL and the SECLABEL in the VMLAN profile. The outcome of the request depends on the SETROPTS options that are in effect. For more information, see “Security Label Authorization Checking” on page 315.

If your installation does not want to use Guest LAN protection in the VMLAN class when the SECLABEL class is active, deactivate the class by issuing:

```
SETROPTS NOCLASSACT(VMLAN)
```

z/VM still calls RACF for MAC authorization, but RACF defers the authorization to z/VM. z/VM will then perform the COUPLE authorization based on definitions in z/VM.

If the SECLABEL class is active and the SETROPTS MLACTIVE(FAILURES) option is in effect, profiles in the VMLAN class are required to contain a SECLABEL. Therefore, authorization requests for a Guest LAN which does not have an assigned SECLABEL will fail. If a VLAN ID-qualified profile does not have a SECLABEL, then the user will not be authorized to that VLAN ID. All the VLAN ID-qualified profiles should have the same SECLABEL as that of the base profile protecting the virtual switch.

Protecting Terminals on VM

There are several methods of controlling the use of terminals connected to your VM system. If you create profiles in the TERMINAL or GTERMINL classes, access to certain terminals is protected. You can also choose options that will prevent access to undefined terminals, limit specific users to specific terminals, and restrict access to the system during certain hours and days of the week.

For a description of authorization checking for terminals, see “Using Security Labels to Control Terminals” on page 184.

Creating Profiles in the TERMINAL and GTERMINL Classes

If you create a profile in the TERMINAL or GTERMINL class, users must have at least READ access authority to the profile in order to use the terminal(s) protected by the profile.

1. To protect a terminal using RACF, create a profile for it using the RDEFINE command. On the command, specify the universal access authority (UACC) you want to assign to the terminal. The profile name consists of a prefix concatenated with the real address of the terminal.

   For example, if the prefix is LOGN, for a terminal whose real address is 110B, you would define it as follows:

   ```
   RDEFINE  TERMINAL  LOGN110B  UACC(NONE)
   ```

   Notes:

   a. If your system has PVM installed, users might be able to enter the system from a protected terminal by issuing the DIAL PVM command, then logging on. The logon occurs at a “logical terminal,” whose address cannot be predicted. You can prevent this by issuing the following command:

      ```
      SETEVENT NODIAL
      ```

      This prevents users from issuing the DIAL PVM command. (Users will still be able to use PVM by logging on, then issuing the PASSTHRU command.)

   b. To determine which address to use when defining a terminal, do one of the following:

      - Go to the terminal itself, and issue the following command before logging on to the system:

        ```
        MESSAGE * Test message
        ```

The message will indicate the address of the terminal unless you are issuing the message from a VTAM terminal, in which case you must use the VTAM LU name to define the terminal. (See the next bulleted item.)

For a discussion of prelogon messages for SETEVENT LIST, refer to *RACF Command Language Reference*.

- If someone is currently logged on to the terminal, and is not using a *dialed* terminal, issue the following command:

  ```
  QUERY userid
  ```

  The system will indicate the user's terminal address which, in the case of a VTAM terminal, will also be the VTAM LU name.

2. Use the PERMIT command to allow users and groups to use the terminal. You must give a user at least READ access authority to the terminal. Otherwise, the user will not be authorized to use the terminal. For example, the following command grants users SMITH and JONES READ access authority to terminal LOGN110B.

   ```
   PERMIT LOGN110B CLASS(TERMINAL) ID(SMITH JONES) ACCESS(READ)
   ```

   Attention

   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 profile.

3. When you are ready to start using the protection defined in the profiles, activate the TERMINAL class. You should also consider activating SETROPTS RACLIST processing for the class. SETROPTS RACLIST processing helps ensure high performance when access authorities are checked. Also, if you are using GTERMINL profiles, you *must* request RACLIST processing for the TERMINAL class. You can do these two actions in one command:

   ```
   SETROPTS  CLASSACT(TERMINAL)  RACLIST(TERMINAL)
   ```

   Note: When you activate the TERMINAL class, RACF also activates the GTERMINL class.

SETROPTS RACLIST Processing on Shared Systems

The RACLIST processing for SETROPTS applies only to the system (VM or MVS) on which you issue the SETROPTS command. If your installation has two or more systems sharing a RACF database, you must issue the SETROPTS command on all systems to have the RACLIST done on all systems. In a multiple RACF service machine environment, you must also issue the SETROPTS command to each service machine that shares the RACF database. (To coordinate the command across multiple RACF service machines, refer to the RAC command information in *RACF Command Language Reference*.)

However, if you do not issue the SETROPTS command with the RACLIST option on a system sharing a RACF database and that system needs to re-IPL, RACLIST is performed for that system when a re-IPL occurs.

Creating a Profile in the GTERMINL Class: If you want to protect several terminals in the same way, but their real addresses do not allow you to create a generic profile, you can create a profile in the GTERMINL class for them. For example, to protect terminals at addresses 14CC, 35AB, and 20EE with one profile, you could create a profile with a name you choose, such as DEPT35:

```
RDEFINE  GTERMINL  DEPT35  UACC(NONE)
         ADDMEM(LOGN14CC LOGN35AB LOGN20EE)
```

To allow group FINANCE to use these terminals, enter the following:

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

Note: After creating or changing a GTERMINL profile, you must request SETROPTS RACLIST processing for the TERMINAL class to make the changes effective on the system.

To protect another terminal, whose real address is 26DD, with the same profile, change the DEPT35 profile as follows:

```
RALTER  GTERMINL  DEPT35  ADDMEM(LOGN26DD)
SETROPTS  RACLIST(TERMINAL)  REFRESH
```

To stop protecting the terminal whose real address is 35AB with this profile, change the DEPT35 profile as follows:

```
RALTER  GTERMINL  DEPT35  DELMEM(LOGN35AB)
SETROPTS  RACLIST(TERMINAL)  REFRESH
```

SETROPTS REFRESH Processing on Shared Systems

The refresh operation for SETROPTS processing applies only to the system (VM or MVS) on which you issue the SETROPTS command. If your installation has two or more systems sharing a RACF database, you must issue the SETROPTS command on all systems to have the refresh done on all systems. In a multiple RACF service machine environment, you must also issue the SETROPTS command to each service machine that shares the RACF database. (To coordinate the command across multiple RACF service machines, refer to the RAC command information in *RACF Command Language Reference*.)

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

Preventing the Use of Undefined Terminals

You can also use RACF to control the use of undefined terminals connected to your system. To control the use of undefined terminals, you must first activate the TERMINAL class as shown above. After the TERMINAL class is active, you can control whether users can log on to undefined terminals by issuing the SETROPTS command with the TERMINAL operand. The TERMINAL operand specifies the universal access authority, either READ or NONE, that RACF associates with undefined terminals on your system.

To allow undefined terminals to be used for logging on, enter the following:

```
SETROPTS  TERMINAL(READ)
```

To prevent undefined terminals from being used for logging on, enter the following:

```
SETROPTS  TERMINAL(NONE)
```

Attention

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.*

Combining the SETROPTS TERMINAL Command with TERMINAL Profiles

If you want to control selected terminals, specify READ. When you specify READ, all users can access all terminals. To control access to selected terminals, define each terminal individually and specify a UACC of NONE. Then create an access list for each terminal containing the user IDs of the users who require access to the terminal.

If you decide that you want to control *all* terminals, specify NONE on the TERMINAL operand of the SETROPTS command. When you specify NONE, only users and groups that you authorize to use a terminal through its access list can use it. (See **Attention** box above.)

Restricting Specific Groups of Users to Specific Terminals

When defining or changing a group profile, you can specify that the group can only log on to those terminals to which the group (or individual users within the group) are specifically authorized. If the group terminal option NOTERMUACC is in effect (note that TERMUACC is the default) for a group on the ADDGROUP or ALTGROUP command, users of the group can use only those terminals to which they are specifically authorized on the access list in the TERMINAL profile protecting the terminal.

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

```
RDEFINE  GTERMINL  PAYTERMS  ADDMEM(LOGN1344 LOGN1345)  UACC(NONE)
```

Give the PAYROLL group READ access:

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

Ensure that the PAYROLL group profile has NOTERMUACC specified:

```
ALTGROUP  PAYROLL  NOTERMUACC
```

This prevents users in group PAYROLL from logging on to another terminal just because the profile protecting that terminal has a UACC of READ.

Note: If the list-of-groups option (SETROPTS GRPLIST) is in effect, RACF uses the TERMUACC/NOTERMUACC option from the user's current connect group, but RACF can grant terminal access through any of the user's connect groups.

Restricting the Times That a Terminal Can Be Used

RACF allows you to limit the use of specific terminals to certain days of the week, and certain hours within each day. To control when the system may be accessed from the terminal, use the WHEN operand on the RDEFINE and RALTER commands for the TERMINAL class. For more information on the time and day-of-week controls, see "Limiting When a User Can Access the System" on page 68, or the command descriptions in *RACF Command Language Reference*.

For example, to allow logons at a terminal only between 7 A.M. and 5 P.M. during the week, specify WHEN(DAYS(WEEKDAYS) TIME(0700:1700)) on the RDEFINE or RALTER command.

Using Security Labels to Control Terminals

If both the TERMINAL and the SECLABEL class are active, RACF checks a user's authority to use a terminal. To use the terminal, the user must log on with a security label that is less than or equal to the security label of the terminal.

You can use this to limit the sensitivity of the data that users can access from the terminal. For example, if you have some terminals that can be accessed easily by many users, you can assign those terminals a low-sensitivity security label, such as SYSLOW. This prevents users from logging onto those terminals to access data that has a security label higher than the terminal's security label.

Note: All of these terminals must be defined to the TERMINAL class.

Additionally, when the SECLABEL class is active and SETROPTS MLACTIVE(FAILURES) has been issued, profiles in the TERMINAL class are required to contain a SECLABEL. Therefore, logon requests for a terminal which does not have an assigned SECLABEL will fail.

Protecting the Use of CP Commands, Diagnose Codes, and Other Functions

You can use RACF to protect certain CP commands, diagnose codes, and other functions with profile names in the VMCMD class, as described in Table 18.

Table 18. VM Command Profiles and What They Protect

VMCMD Profile Name	What It Protects
STORE.C	STORE HOST command
TRSOURCE	TRSOURCE command
DIAG0E4	DIAGNOSE code X'E4'
XAUTOLOG.userid	XAUTOLOG command (issued by a class G user)
DIAG0A0.HRTSTORE	DIAGNOSE A0 Subcode X'34'
DIAG0A0.QUERYSEC	DIAGNOSE A0 Subcode X'30'
DIAG0A0.VALIDATE	DIAGNOSE A0 Subcode X'04'
RAC	RAC command processor and RACF ISPF panels
RACF	RACF command session and RACFISPF

Some of these events are controllable through the use of a VMXEVENT profile. If you have activated a VMXEVENT profile that turns off control for a particular event, RACF will not be called—even if a profile exists in the VMCMD class.

Security Label Considerations for Profiles in the VMCMD Class

If a security label exists in the VMCMD profile protecting the resource, and the SECLABEL class is active, RACF compares the security label of the user with the security label contained in the profile, and grants access if the security label of the user is equal to or higher than the security label contained in the profile.

Protecting the STORE.C, TRSOURCE, and DIAG0E4 Events

The VM event names STORE.C, TRSOURCE, and DIAG0E4 refer to the STORE HOST command, TRSOURCE command, and DIAGNOSE code X'E4', respectively. VM has restrictions on the use of these events (for example, STORE HOST is limited to privilege class C users only), but you can further restrict access to these events with a VMCMD profile for each event.

1. Define the command as a profile in the VMCMD class and specify a UACC of NONE:

   ```
   RDEFINE  VMCMD  command_name  UACC(NONE)
   ```

2. Allow the appropriate users or groups to use the command by giving them READ access:

   ```
   PERMIT command_name CLASS(VMCMD) ID(user or group) ACCESS(READ)
   ```

   Note that if you do not give users an access authority of READ when creating the profile's access list, the users will not be able to use the command.

3. If the VMCMD class is not already active, use the SETROPTS command to activate it:

   ```
   SETROPTS CLASSACT(VMCMD)
   ```

4. Make sure protection for the respective command name is active. (See “Controlling Authorization Checking for VM Events” on page 152.) It is on by default if an installation has not changed the setting in the currently active VMXEVENT profile by issuing, for example:

   ```
   RDEFINE VMXEVENT profile-name ADDMEM(command_name/NOCTL)
   ```

   If protection of the command name is not currently active, activate it by issuing:

   ```
   RALTER VMXEVENT profile-name DELMEM(command_name/NOCTL)
   SETEVENT REFRESH profile-name
   ```

SYSSEC Considerations for STORE.C, TRSOURCE, and DIAG0E4

The SYSSEC macro, coded in the RACF module HCPRWA, can affect the final outcome of resource requests in the VMCMD class for the STORE.C, TRSOURCE and DIAG0E4 events. The SYSSEC macro defines the relationship between RACF's response to a resource access request and the final disposition of that request by VM. This relationship changes as you change the SYSSEC statement in HCPRWA.

For example, when a user issues the TRSOURCE command, there may be no RACF profile to protect the TRSOURCE command. SYSSEC can be coded so that for this case, RACF will do one of the following:

- Allow access

- Disallow access
- Defer the access decision to VM

You should know how SYSSEC is affecting your VMCMD requests. See *RACF Macros and Interfaces* for more information.

Protecting the XAUTOLOG.G Command

For virtual machines that are logged on with the XAUTOLOG command, you can control which other privilege class G virtual machines can do the XAUTOLOG. To do this, create profiles named XAUTOLOG.*userid*, where *userid* is the virtual machine being logged on with the XAUTOLOG command. Give READ access to the privilege class G user ID that will be issuing the XAUTOLOG command.

For example, for USER1 to issue the command XAUTOLOG USER2, USER1 must have at least READ access authority to the VMCMD profile XAUTOLOG.USER2. No password is required to be entered.

1. Define the XAUTOLOG user ID as a profile in the VMCMD class and specify a UACC of NONE:

   ```
   RDEFINE  VMCMD  XAUTOLOG.userid  UACC(NONE)
   ```

2. Allow the appropriate users or groups to use the command by giving them READ access:

   ```
   PERMIT XAUTOLOG.userid CLASS(VMCMD) ID(user or group) ACCESS(READ)
   ```

   Note that if you do not give users an access authority of READ when creating the profile's access list, the users will not be able to use the command.

3. If the VMCMD class is not already active, use the SETROPTS command to activate it:

   ```
   SETROPTS CLASSACT(VMCMD)
   ```

For information on the XAUTOLOG command when the SECLABEL class is active, see “Security Label Considerations for Profiles in the VMCMD Class” on page 185 and “LOGON, AUTOLOG, and XAUTOLOG Commands” on page 190.

SYSSEC Considerations for XAUTOLOG.G

The SYSSEC macro, coded in the RACF module HCPRWA, can affect the final outcome of resource requests in the VMCMD class for the XAUTOLOG.G command. The SYSSEC macro defines the relationship between RACF's response to a resource access request and the final disposition of that request by VM. This relationship changes as you change the SYSSEC statement in HCPRWA.

For example, a user may issue the XAUTOLOG USER01 command, and there is no RACF profile named XAUTOLOG.USER01 in the VMCMD class. SYSSEC can be coded so that for this case, RACF will do one of the following:

- Allow access
- Disallow access
- Defer the access decision to VM

You should know how SYSSEC is affecting your VMCMD requests. See *RACF Macros and Interfaces* for more information.

Protecting the DIAGNOSE X'A0' Subcodes

RACF provides four subcodes of DIAGNOSE X'A0' that can be used by application programs. The use of three of the subcodes (X'04', X'30', and X'34') can be protected using RACF profiles.

RACF profile protection will be in effect and privilege class checking will be ignored for these three DIAGNOSE X'A0' subcodes only if all of the following conditions are met:

- Control for DIAG0A0 is turned on.
- A VMCMD class profile has been defined for the subcode being issued.
- The VMCMD class is active.

If any one or more of the above conditions is not met, privilege class checking is enforced.

Table 19. DIAGNOSE A0 Subcodes

Subcode	Description	Privilege Class	VMCMD Class Profile Name
X'04'	Use subcode X'04' to verify a user and validate the user's password.	One or more of: A B C D E F	DIAG0A0.VALIDATE
X'30'	To query the current SECLABEL of your own user ID.	ANY	None (ANY privilege class **cannot** be protected)
	To query the current SECLABEL of a different user ID.	One or both of: • A • B	DIAG0A0.QUERYSEC
X'34'	Use subcode X'34' to update the human-readable-label to SECLABEL correlation table ("HR table") in CP.	One or both of: A B	DIAG0A0.HRTSTORE
X'38'	Use subcode X'38' to obtain the size of, or a copy of the human-readable-label to SECLABEL correlation table ("HR table").	ANY	None (ANY privilege class **cannot** be protected)

You cannot use the VM User Class Restructure (UCR) function to change the privilege class of DIAGNOSE X'A0' because DIAGNOSE X'A0' is defined to VM as a privilege class ANY diagnose code. Also, if you allow a user ID a specific privilege class, the user ID can perform all tasks at that privilege class level; you may not want to grant the user that level of privilege. For these reasons, it is recommended that you give user IDs access to the VMCMD class profiles rather than give them the required privilege class.

Note: When DIAGNOSE X'A0' control is in effect, regardless of profile existence, RACF is called each time **any** of the protectable subcodes is used; performance impacts are related to the frequency with which the user ID uses the subcode.

Setting Up RACF Protection for a Subcode

To set up protection for one of the three subcodes:

1. Define the subcode as a profile in the VMCMD class and specify a UACC of NONE:

   ```
   RDEFINE  VMCMD  DIAG0A0.profile_name  UACC(NONE)
   ```

2. Allow the appropriate users or groups to use the subcode by giving them READ access to the appropriate profile:

   ```
   PERMIT DIAG0A0.profile_name CLASS(VMCMD) ID(user
   or group) ACCESS(READ)
   ```

   Note that if you do not give users an access authority of at least READ when creating the access list, the users will not be able to use the subcode.

3. If the VMCMD class is not already active, use the SETROPTS command to activate it:

   ```
   SETROPTS CLASSACT(VMCMD)
   ```

4. Make sure protection of DIAG0A0 is active. It is active by default if an installation has not changed the setting in the currently active VMXEVENT profile by issuing, for example:

   ```
   RDEFINE VMXEVENT EVENTS1 ADDMEM(DIAG0A0/NOCTL)
   ```

   If protection of DIAG0A0 is not currently active, activate it by issuing:

   ```
   RALTER VMXEVENT EVENTS1 DELMEM(DIAG0A0/NOCTL)
   SETEVENT REFRESH EVENTS1
   ```

SYSSEC Considerations for DIAG0A0

The SYSSEC macro, coded in the RACF module HCPRWA, can affect the final outcome of resource requests in the VMCMD class for the DIAG0A0 event. The SYSSEC macro defines the relationship between RACF's response to a resource access request and the final disposition of that request. This relationship changes as you change the SYSSEC statement in HCPRWA.

For example, a user may issue DIAGNOSE X'A0' subcode X'04', and there is no RACF profile named DIAG0A0.VALIDATE in the VMCMD class. SYSSEC can be coded so that for this case, RACF will do one of the following:

- Allow access
- Disallow access
- Defer the access decision to privilege class checking

You should know how SYSSEC is affecting your VMCMD requests. See *RACF Macros and Interfaces* for more information.

Protecting the RAC Command Processor and the RACF Command Session

Two profiles in the VMCMD class, RACF and RAC, protect access to the RACF command session, RACFISPF, the RAC command processor, and the RACF panels. Use these two profiles to limit or to allow access on your system:

- Use the RACF profile to protect the use of the RACF command session and RACFISPF.

- Use the RAC profile to protect the use of the RAC command processor and the RACF panels.

If you have both the RACF and the RAC profiles set to a UACC of NONE, and if you have not permitted anyone through an access list, no one will be able to use any of the RACF commands or panels. An absence of one of the profiles means that everyone is permitted to use what that profile was protecting. For each of the profiles, RACF and RAC, READ access permits users to use the commands or panels.

Note: Your installation may require users to supply their logon passwords to enter the RACF command session and RACFISPF. If users choose to change their passwords at this time, and are then denied access to the RACF command session because of a RACF profile in place, their password change is still in effect.

Examples:

1. Create a profile in the VMCMD class:

   ```
   RDEFINE  VMCMD  RACF  UACC(NONE)
   ```

   or

   ```
   RDEFINE  VMCMD  RAC   UACC(NONE)
   ```

   Note: If the RACF or RAC profiles do not exist, all users can use a RACF command session and RACFISPF, or RAC and the panels, respectively.

2. Allow selected users to use the respective functions:

   ```
   PERMIT  RACF  CLASS(VMCMD) ID(user or group) ACCESS(READ)
   ```

   or

   ```
   PERMIT  RAC  CLASS(VMCMD) ID(user or group) ACCESS(READ)
   ```

   Note: Users of the BLKUPD utility must use the RACF command session. Therefore you should give them access to this profile.

3. Activate the VMCMD class:

   ```
   SETROPTS CLASSACT(VMCMD)
   ```

Activating the Security Label Class on VM

This section discusses the SECLABEL considerations that are specific for RACF on VM. For general information about SECLABELs with RACF, see, Chapter 7, “Security Classification of Users and Data” on page 111.

When the SECLABEL class becomes active, RACF indicates to VM that mandatory access control (MAC) has been enabled. VM then unconditionally calls RACF to determine MAC authorization for the subset of VM events which require MAC authorization.

When the SECLABEL class is active, VM does not call for MAC authorization *until* after one authorization or audit call is made to RACF. Once any call from VM to RACF has been made for a VM event, VM begins calling RACF for MAC authorization. In other words, VM is not notified when the SECLABEL class is active. Instead, VM is notified when VM calls RACF next. For example, to get VM to call RACF at least once, you could do any of the following:

- Log on to any user (VM always calls RACF for logon authorization)
- Issue a LINK command (if you do not have a LINK/NOCTL member in an active VMXEVENT profile)
- Issue a command for which you have turned on VMXEVENT auditing.

While the SECLABEL class is active, control cannot be turned off selectively for the VM events that require MAC authorization. Unlike using VMXEVENT profiles and the SETEVENT command to control which VM events RACF is called to protect, when MAC authorization is in effect you cannot turn on VM calls for MAC authorization for some events and others off.

Table 20 shows where to find discussions of how security label authorizations affect particular commands and classes.

Table 20. Where to Find Specific Security Label Information

Resource	Chapter Reference
LINK and MDISK Commands	"Security Label Checking for LINK and MDISK Events" on page 168
Unit Record Devices	"Security Label Considerations for Unit Record Devices" on page 171
RSCS Nodes	"Security Label Considerations for RSCS Nodes" on page 172
Alternate User IDs	"Security Label Considerations for Alternate User IDs" on page 174
VMSEGMT	"Security Label Considerations in the VMSEGMT Class" on page 176
Guest LAN	"Security Label considerations for Guest LANs" on page 179
Terminals	"Using Security Labels to Control Terminals" on page 184
Profiles in the VMCMD Class	"Security Label Considerations for Profiles in the VMCMD Class" on page 185
SFS files and directories	"Security Label Considerations for SFS Files and Directories" on page 270

The following sections discuss security label considerations for other commands that are affected by MAC authorization.

LOGON, AUTOLOG, and XAUTOLOG Commands

VM always calls RACF for the LOGON, AUTOLOG, and XAUTOLOG commands. When the SECLABEL class is active and the users have a default SECLABEL defined in their user profiles, RACF ensures these users are authorized to the SECLABEL before allowing them to log on or be logged on.

When users are logged on using AUTOLOG or XAUTOLOG, they are assigned the default SECLABEL from their user profiles. When users log on with the LOGON command, a SECLABEL may be specified on the LOGON command; this SECLABEL is then used for authorization checking. If no SECLABEL is specified, the default SECLABEL for these users is used for authorization checking. For more information, see *RACF General User's Guide*.

Additionally, when the SECLABEL and the VMMAC classes are active, an authorization check is made in the VMMAC class for the XAUTOLOG and AUTOLOG commands and when a user logs on to a logical device. See "Protecting VM Events with the VMMAC Class" on page 192 for more information.

User IDs Autologged During System Initialization

Several user IDs are automatically started by the VM operating system during system initialization. These user IDs include the following: AUTOLOG1, OPERACCT, OPEREREP, OPERSYMP, and OPERATOR. The autologged user IDs are tailorable using a system configuration file called SYSTEM CONFIG. RACF is not active until after these users have been logged on.

When these user IDs are logged on as part of VM initialization, VM assigns a SECLABEL of SYSHIGH to these users. If the SECLABEL class is active, for these user IDs to function properly once RACF is initialized, the system administrator should define SYSHIGH as the default SECLABEL for these users and permit the users to use the SYSHIGH security label.

For more information on SYSTEM CONFIG, see *z/VM Planning and Administration.*

Protecting a VM System Printer

When the SECLABEL class is active, VM calls RACF when a VM system printer is started. The printer operator (or issuer of the START command) specifies the security label of the data to be printed on the printer.

To start the printer, the security label of the data to be printed on the printer must be less than or equal to the security label of the printer. The security label of the printer is defined in the WRITER class profile for that printer.

In preparation for this authorization checking, you must do the following:

1. Define a profile in the WRITER class for each VM system printer, assigning a SECLABEL to the profile that represents the highest security level of data that can be printed on that printer. The printer name is the real device address of the printer, including leading zeros.

   For example, if the printer at device 0008 is allowed to print data up to and including the SECLABEL SECRET, define the profile as follows:

   ```
   RDEFINE  WRITER  0008  SECLABEL(SECRET)  UACC(NONE)
   ```

2. Authorize the appropriate users to be able to start the printer. READ access is required.

   ```
   PERMIT 0008 CLASS(WRITER) ID(USER01) ACCESS(READ)
   ```

3. Activate the WRITER class.

   ```
   SETROPTS CLASSACT(WRITER)
   ```

If your installation wants to use SECLABEL checking, but does not want to use SECLABEL checking for the WRITER class, deactivate the class by issuing:

```
SETROPTS NOCLASSACT(WRITER)
```

VM still calls for MAC authorization, but RACF defers the authorization to VM. The use of SECLABEL *NONE* is not allowed; it is assigned by VM when the printer is initialized. For restrictions on the use of a SECLABEL of NONE with VM system printers, see “Security Label NONE” on page 120.

Spool File Considerations

When the SECLABEL class is active, each spool file is assigned a security label when it is created. The security label assigned to the spool file is the security label at which the spool file creator is logged on. The security label is carried with the spool file instead of being stored in the RACF database. See also “Protecting VM Events with the VMMAC Class” for more security label considerations for VM events that manipulate spool files.

Printing a Spool File

When the SECLABEL class is active, VM selects spool files to print based on the security label at which a printer is started (see “Protecting a VM System Printer” on page 191). For example, if the printer is started at the security label SECRET, only spool files with a security label of SECRET are allowed to print on that printer.

Once VM selects a file to be printed, VM calls RACF to see if the user is authorized to print the file. (The VM event for this call is UTLPRINT.) RACF examines the security label of the print spool file and determines if the user who printed the file is permitted to use the security label of the print spool file. This security label authorization is not related to the security label at which the user is currently logged on, because the user may be logged off when the file is actually printed.

Based on the security label authorization, if a user is not authorized to print a file, VM places the print file in SYSTEM HOLD status and sends a message to the system operator. The user is not notified of the print failure.

For example, if a user is authorized (through the PERMIT command) to use security labels SECRET and NOSECRET, but tries to print a spool file that is labeled SYSLOW, VM does not print the file. Instead, the file is placed in SYSTEM HOLD status and the system operator is notified.

Protecting VM Events with the VMMAC Class

When the SECLABEL class is active, the VM events requiring MAC authorization can be protected with the VMMAC class. To activate the VMMAC class, issue:

```
SETROPTS CLASSACT(VMMAC)
```

No profiles exist for this class. Instead, when authorization checking is performed for this class, RACF checks if the subject of the command (usually the command issuer) has the appropriate authority to the SECLABEL of the target resource. For details on the kind of SECLABEL authorization performed for each event, see Table 21 on page 193.

The outcome of each request depends on the SETROPTS options that are in effect. For more information, see “Security Label Authorization Checking” on page 315.

If your installation wants to use SECLABEL checking, but does not want to use SECLABEL checking for the VMMAC class, deactivate it by issuing:

```
SETROPTS NOCLASSACT(VMMAC)
```

VM still calls RACF for MAC authorization, but RACF defers the authorization to VM. VM treats this as a successful MAC authorization for these events.

Table 21 (Page 1 of 2). Security Label Authorizations in the VMMAC Class

VM Event	Privilege Class	RACF Class	Type of SECLABEL Author-ization	SECLABELs Compared
AUTOLOG	AB	USER and VMMAC	W/O	Issuer to autologee
CHANGE	G	VMMAC	W/O	Issuer to spool file
COUPLE	G	VMMAC	R/W	Issuer to target user (for CTCA)
LOGON (to LDEV)	ANY	USER and VMMAC	R/W	User logging on to creator of logical device
MESSAGE	ANY	VMMAC	W/O	Issuer to receiver
MSGNOH	B	VMMAC	W/O	Issuer to receiver
QUERY READER	G	VMMAC	R/O	Issuer to spool file
QUERY PRINTER	G	VMMAC	R/O	Issuer to spool file
QUERY PUNCH	G	VMMAC	R/O	Issuer to spool file
QUERY TAG FILE	G	VMMAC	R/O	Issuer to spool file
QUERY TRFILES	ACDEG	VMMAC	R/O	Issuer to spool file
SMSG	G	VMMAC	W/O	Issuer to receiver
TAG FILE	G	VMMAC and VMNODE	W/O	Issuer to spool file
TAG QUERY FILE	G	VMMAC	R/O	Issuer to spool file
WARNING	ABC	VMMAC	W/O	Issuer to receiver
XAUTOLOG	AB	USER and VMMAC	W/O	Issuer to autologgee
XAUTOLOG	AB	USER and VMMAC	R/W	Autologgee to creator of logical device
XAUTOLOG	G	USER, VMMAC, and VMCMD	W/O	Issuer to autologee
DIAG014 (subcodes X'0004' X'0008' X'0FFE' X'0FFF')	ANY	VMMAC	R/O	Issuer to spool file
DIAG068 (subcode X'000A' VMCF Identify)	ANY	VMMAC	R/W	Issuer to receiver
DIAG068 (subcode X'0002' VMCF Send)	ANY	VMMAC	W/O	Issuer to receiver
DIAG068 (subcode X'0003' VMCF Send and Receive)	ANY	VMMAC	R/W	Issuer to receiver
DIAG068 (subcode X'0004' VMCF Sendx)	ANY	VMMAC	W/O	Issuer to receiver
DIAG068 (subcode X'0005' VMCF Receive)	ANY	VMMAC	R/O	Issuer to receiver
DIAG068 (subcode X'0007' VMCF Reply)	ANY	VMMAC	W/O	Issuer to receiver
DIAG0B8 (subcode X'0000')	ANY	VMMAC	R/O	Issuer to spool file
DIAG0B8 (subcode X'0004')	ANY	VMMAC	W/O	Issuer to spool file
DIAG0BC	ANY	VMMAC	R/O	Issuer to spool file
DIAG23C (subcode X'03')	ANY	VMMAC	R/O or R/W	User being permitted to issuer
APPCCON	n/a	VMMAC	R/W	Issuer to target user
IUCVCON	n/a	VMMAC	R/W	Issuer to target user

Table 21 (Page 2 of 2). Security Label Authorizations in the VMMAC Class

VM Event	Privilege Class	RACF Class	Type of SECLABEL Author-ization	SECLABELs Compared
SDF_OPEN[1]	n/a	VMMAC	R/O	Issuer to spool file
SPF_OPEN[1]	n/a	VMMAC	R/O	Issuer to spool file

Note: [1] An SDF_OPEN or SPF_OPEN can be issued by other VM events like DIAG0E0, DIAG034, or DIAG014. This can cause a SECLABEL authorization check for these other events.

Use of ACIGROUP Control Statements

If a user's directory entry has an ACIGROUP control statement, the following considerations apply:

- If the SETROPTS GRPLIST option is in effect, the group specified on the ACIGROUP control statement for a user must be a RACF-defined group of which the user is a member.
- If the SETROPTS GRPLIST option is *not* in effect, the group specified on the ACIGROUP control statement for a user must be the user's default connect group.
- Profiles for the user's minidisks and virtual unit record devices are prefixed with the user's ACIGROUP name. For example, if user SUE's directory entry has an ACI group of GRP1, her 191 disk is protected by the following profile:

  ```
  RDEFINE  VMMDISK  GRP1.SUE.191 UACC(NONE)
  ```

  and her virtual reader, printer, and punch are protected by:

  ```
  RDEFINE  VMRDR  GRP1.SUE UACC(NONE)
  ```

- You can use the RACGROUP EXEC to determine a user's access control group:

  ```
  RACGROUP userid
  ```

Remote APPC Connections and Their Effects on RACF for VM

Using the APPCPWVL Event To Verify Passwords On APPC CONNECT

When SECURITY(PGM) is specified on an APPC CONNECT to the local system, a user ID and password combination is supplied for verification. If the APPCPWVL event is being controlled in your currently active VMXEVENT profile, VM will call RACF to verify that the password is correct for the specified user ID. If APPCPWVL is not being controlled, VM will verify the user ID and password combination against information in the CP directory.

When RACF is called to check the password, the user's count of successive incorrect passwords will be incremented if the password is not correct. If the limit established on SETROPTS(PASSWORD(REVOKE)) is reached, the user will be revoked by RACF on the local system.

Considerations When the SECLABEL and VMMAC Classes Are Active

When RACF is active, VM may call RACF for an APPC connection (VM event APPCCON), which originates from a remote system. If the SECLABEL and VMMAC classes are active, the remote user ID that originated the connection must be defined to RACF. It must also have a user profile in the RACF database.

For example, if user SYSAUSER on System A issued an APPC connection to SYSBUSER on System B and the SECLABEL and VMMAC classes were active, the user ID SYSAUSER must be defined in the RACF database on System B.

Chapter 11. Using the Secured Signon Function

If your installation includes workstations and client machines that are operating in a client/server environment, you may want to use the RACF secured signon function to provide enhanced security across a network. The secured signon function provides an alternative to the RACF password called a PassTicket, which allows workstations and client machines to communicate with a host without using a RACF password.

The secured signon function removes the need to send RACF passwords across the network and allows you to move the user authentication part of signing on to a host from RACF to another product or function. End users of an application can use the PassTicket to authenticate their user IDs and log on to computer systems that contain RACF.

This section describes the PassTicket and how to set up the secured signon environment. It includes information about:

- Activating the PTKTDATA class
- Defining profiles in the PTKTDATA class
- The process RACF uses to validate a password or PassTicket
- Enabling the use of PassTickets

For information about the programming that is needed for an application to generate a PassTicket, see *RACF System Programmer's Guide*.

The RACF PassTicket

The RACF PassTicket is a *one-time-only*[1] password that is generated by a requesting product or function. It is an alternative to the RACF password that removes the need to send RACF passwords across the network in clear text. It makes it possible to move the authentication of a mainframe application user ID from RACF to another authorized function executing on the host system or to the workstation local area network (LAN) environment.

Activating the PTKTDATA Class

Before you can use the secured signon function, you must activate the PTKTDATA class. The PTKTDATA class is the class to which all profiles that contain PassTicket information are defined. To activate the class and the function, enter:

```
SETROPTS CLASSACT(PTKTDATA)
SETROPTS RACLIST(PTKTDATA)
```

After you activate the PTKTDATA class, you can define the necessary profiles.

Note: After you define or change the profiles, you need to refresh the class by entering `SETROPTS RACLIST (PTKTDATA) REFRESH`.

Defining Profiles in the PTKTDATA Class

For each application that users can gain access to with the PassTicket, you must create at least one profile in the PTKTDATA class. The profile associates a secret secured signon application key with a particular application on a particular system.

To define the profile, use the RDEFINE command:

```
RDEFINE PTKTDATA profile_name
        SSIGNON(key_description)
        UACC(access_authority)
```

where:

PTKTDATA
specifies the PassTicket Key class.

profile_name
is the name of the profile (see “Determining Profile Names” on page 199).

For the PTKTDATA class, the profile must be a discrete profile. Because each application must be uniquely defined, you cannot specify a generic profile in the PTKTDATA class. If you specify a generic profile, it is ignored during PassTicket processing for the application, and PassTickets cannot be used to authenticate users for that application.

[1] Because it only gives one user access to a specific application for approximately 10 minutes, a RACF PassTicket is resistant to reuse. For most applications, once a particular PassTicket is used, the same user cannot use it again for the same application during the same 10-minute interval.

By keeping a copy of all used valid PassTickets for the duration of the 10-minute interval during which they might possibly be used again, RACF provides another level of protection against reuse. For performance reasons, RACF uses main memory for this storage. If an application can run on more than one computer with individual memory at the same time, this level of reuse protection might not be available.

key_description
defines the secured signon application key and specifies the method RACF is to use to protect it in the RACF database on the host. You can specify either masking or encryption for the method (see “Protecting the Secured Signon Application Keys” on page 200).

Secured signon keys are 64-bit Data Encryption Standard (DES) keys. With DES, 8 of the 64 bits are reserved for use as parity bits, so those 8 bits are not part of the 56-bit key. In hexadecimal notation, the DES parity bits are: `X'0101 0101 0101 0101'`. Any two 64-bit keys are equivalent DES keys if their only difference is in one or more of these parity bits.

access_authority
is the universal access authority to be associated with the resource protected by this profile. By default, the UACC is NONE for the PTKTDATA class.

After a profile in the PTKTDATA class has been created, you can change it with the RALTER command, which is similar in syntax to the RDEFINE command:

```
RALTER PTKTDATA profile_name
       SSIGNON(key_description)
       UACC(access_authority)
```

Determining Profile Names

Depending on the application, the secured signon function uses a specific method for determining profile names in the PTKTDATA class. For information on using the RACF secured signon function with CICS, IMS, APPC, TSO and MVS applications such as MVS batch, see the RACF Version 2 library. Information follows for using RACF secured signon for VM/CP logons.

VM/CP Logon

1 Ask the system programmer to determine the system ID by examining the CPU-ID (System ID) field in the RACF SMF CONTROL file.

2 Create the profile name to represent VM/CP to the PTKTDATA class by prefacing the CPU-ID field with the characters `VM`.

Attention

If the CPU-ID field contains blanks or characters that are not alphanumeric, they cannot be specified in the profile name. For example:

> If the CPU-ID field contains VM 7, you must specify the profile defined to the PTKTDATA class as VMVM7.

Note: Because each application must be uniquely defined, you cannot use generic profiles to specify profiles to the PTKTDATA class. If you specify a generic profile, it is ignored during PassTicket processing for the application, and PassTickets cannot be used to authenticate users for that application.

Protecting the Secured Signon Application Keys

When you define the secured signon application keys, RACF either masks or encrypts each key. If the system has a cryptographic product installed and available, you can encrypt the application keys for added protection. See “Masking the Secured Signon Application Key” or “Encrypting the Secured Signon Application Key” for more information

Note: Be sure the universal access authority (UACC) of the RACF data base is NONE. This prevents unauthorized users from listing or copying the RACF data set that contains the sensitive RACF secured signon application keys.

Masking the Secured Signon Application Key

If the system using the secured signon function does not use a cryptographic product, RACF masks the key with a proprietary masking algorithm when you define or alter it. The masking algorithm that masks the application keys while they reside on the RACF data base is an IBM proprietary algorithm. It resides within the RACF object code portion of the RACF program product and is designed to provide protection against casual viewing of the secured signon masked keys. The algorithm is *not* a cryptographic algorithm and cannot provide the level of security for the secured signon keys that the use of cryptography can provide.

To mask the application key when you define or alter it, use the SSIGNON operand and KEYMASKED suboperand with the RDEFINE or RALTER command.

Encrypting the Secured Signon Application Key

You can encrypt the secured signon application keys when both of the following are true:

1. All MVS systems you are using to generate or evaluate the PassTicket are running on MVS Version 3 or later.
2. A common cryptographic architecture (CCA) cryptographic product is installed on the systems where the secured signon function is installed.

   Note: RACF for VM does not support the use of cryptographic products.

Using a cryptographic product ensures maximum possible security for the RACF secured signon keys.

With a cryptographic product, RACF can store the keys on the database in a form in which they are encrypted under the cryptographic product's master key. RACF uses the functions of the cryptographic product to be sure the encrypted keys do not exist in clear-text form within system main storage for RACF processing, except when they are being defined. Therefore, if a system storage dump occurs, they are not exposed in the dump.

Sharing a RACF Database

- If you want to encrypt the secured signon application keys when a cryptographic product is installed on one or more of the systems that share a RACF database, but not *all* the systems, you must ensure that the applications requiring the encrypted keys run *only* on the systems on which the cryptographic product is installed.
- If the possibility exists that the application may execute on a system that does *not* have a cryptographic product installed, you must mask the secured signon application keys.

When the Profile Definitions Are Complete

After you define the PTKTDATA-class profile for the application program that is to generate a PassTicket, the program can be installed and used.

For information on how to code an application program to generate a PassTicket, see *RACF Macros and Interfaces*.

How RACF Processes the Password or PassTicket

To validate a password or PassTicket, RACF does the following:

1. Determines whether the value in the password field is the RACF password for the user ID.
   - If it is the RACF password, the validation is complete.
   - If it is not the RACF password, processing continues.
2. Determines whether a secured signon application profile has been defined for the application in the PTKTDATA class.
   - If a profile has not been defined, the user receives a message from the application[2] indicating that the password is not valid.
   - If the application is defined to the PTKTDATA class, processing continues.
3. Evaluates the value entered in the password field. The evaluation determines whether:
   - The value is a PassTicket consistent with this user ID, application, and time range.
   - It has been used previously on this computer system for this user ID, application, and time range.

   Time Considerations:

   A PassTicket is considered to be within the valid time range when the time of generation, with respect to the clock on the generating computer, is within plus or minus 10 minutes of the time of evaluation, with respect to the clock on the evaluating computer.

[2] RACF sends a message to the appropriate log, and to the security console. The application rejects the logon request the same way it rejects an incorrect password. The text of the message the user receives depends on the application.

Be sure that your systems which generate and evaluate PassTickets use clock values that are within that time range. RACF uses the value stored for coordinated universal time (UTC), formerly called Greenwich mean time (GMT), in the algorithms that process PassTickets.

One way to ensure that reasonably synchronized values are used, is to set UTC in the GMT value of the time of day (TOD) clock and to set a similar value in each of the other systems with which RACF shares PassTicket information. For example, you can still use the MVS or VM local time for local timestamp information. Resetting the local time does not affect the GMT value kept in the TOD clock.

Attention

Before setting the TOD clock's GMT value to UTC, make sure that the subsystems and applications you use are not affected.

For more information on setting clocks, see:

- *z/VM CP Planning and Administration*
- *z/VM Virtual Machine Operation*
- *MVS/ESA Initialization and Tuning Reference*
- *MVS/ESA System Commands Reference*.

For more information on the RACF secured signon algorithms, see *RACF Macros and Interfaces*.

- If the value was used before, the user receives a message from the application[3] indicating that the password is not valid.
- If the value was not used before, the PassTicket is considered valid and processing continues.

Determines whether the value is a valid PassTicket.

- If the PassTicket is valid, RACF gives the user access to the desired application.
- If the value is not valid, the host application sends a message[4] to the user indicating that the password is not valid.

Note: If the secured signon application key is encrypted, the cryptographic product must be active when RACF tries to authenticate the PassTicket. If it is not active, RACF cannot validate the PassTicket. The resulting message indicates that the logon attempt failed.

Enabling the Use of PassTickets

To enable RACF to validate PassTickets, the RACF administrator must have:

- Activated the PTKTDATA class.
- Defined a secured signon application key for each application in a profile in the PTKTDATA class.

[3] RACF sends a message to the appropriate log, and to the security console. The application rejects the logon request the same way it rejects an incorrect password. The text of the message the user receives depends on the application.

[4] See the previous footnote.

- Issued the SETROPTS RACLIST(PTKTDATA) command.

As a result, the RACF database contains all the information necessary to validate PassTickets for each application that has a PTKTDATA class profile defined.

Verifying the Secured Signon Environment

After activating the secured signon environment for each application, you should verify the environment. To do this, access the application using the generated PassTicket from a user ID that is able to access that application. This verifies that:

- The application profile and the secured signon application key have been implemented correctly.
- The secured signon environment is active.

Preventing Errors

The following checklist describes the errors that may cause a PassTicket to fail. To prevent these errors from occurring:

1. Read the list before you use the PassTicket.
2. Review your process to ensure that you have entered all of the information correctly.
3. Verify the information by using the procedures described in "Verifying the Secured Signon Environment."

Use this checklist to prevent or correct errors:

__ The PTKTDATA class is activated.

__ You issued the SETROPTS RACLIST(PTKTDATA) command.

__ You issued the SETROPTS RACLIST(PTKTDATA) REFRESH command after defining the profile.

__ A PTKTDATA class profile exists for the application.

__ You issued the RDEFINE command correctly.

Figure 18. Problem Prevention Checklist

Even if you have followed the proper procedures, it is still possible to receive a message stating that a password is incorrect and be denied access to the application. This can occur if:

- The PassTicket was used previously for this user, application, and time range.

  In this case, RACF generates an SMF record that logs an attempt to replay a PassTicket.

- The GMT clock on the evaluating computer is outside the valid time range for the PassTicket.

  This can be caused by one of the following:

  – The GMT clock on the generating computer and the clock on the evaluating computer are not reasonably synchronized.

  – The PassTicket was not used within approximately ten minutes of being generated.

- The system clock on the evaluating computer may not be set correctly in relation to GMT. See Time Considerations on page 201 for more information.

Chapter 12. RACF Controls on VM

Controlling Resources With the FACILITY Class

Planning for Profiles in the FACILITY Class

The FACILITY class can be used for a wide variety of purposes depending on the products installed on your system. If the FACILITY class is active, users might need to have access to particular profiles to perform specific tasks. For example, READ or UPDATE access to profile ICHCONN allows a user to issue certain RACROUTE requests on VM. There are many other profiles used by many products.

You will probably activate the FACILITY class for the first such profile that is required on your system. You will probably create FACILITY profiles as needed to control who can use a number of processes on your system.

It is also recommended that you activate SETROPTS RACLIST processing for the FACILITY general resource class. When you activate this function, you improve performance because I/O to the RACF database is reduced. For a complete description of this function, see "SETROPTS RACLIST Processing" on page 227.

```
SETROPTS  RACLIST(FACILITY)
```

Note: If you activate SETROPTS RACLIST processing for the FACILITY class, any time you make a change to a FACILITY profile, you must also refresh SETROPTS RACLIST processing for the FACILITY class for the change to take effect.

```
SETROPTS  RACLIST(FACILITY)  REFRESH
```

Delegating Authority to Profiles in the FACILITY Class

You can use several methods to allow another user, such as a tape librarian or storage administrator, to work with profiles in the FACILITY class:

- Assign the user as OWNER of all the FACILITY profiles used by the function.
- Create a group representing the function and make the user group-SPECIAL within the group. Then assign the group as OWNER of the FACILITY profiles used by the group.
- If the SETROPTS GENERICOWNER option is in effect, give the user CLAUTH(FACILITY), create a "top" generic profile to which the user is assigned as OWNER. The SETROPTS GENERICOWNER option limits this user to creating FACILITY profiles that are more specific than the "top" generic profile.

  For more information about the GENERICOWNER option, see "Restricting the Creation of General Resource Profiles (GENERICOWNER Option)" on page 220.

Controlling Which Virtual Machines Can Issue RACROUTE Requests

You can use RACF to control which virtual machines can issue RACROUTE requests. This protection applies to all RACROUTE requests that specify RELEASE=1.9 or any later release. (This protection does not apply to RACROUTE requests issued within the RACF service machine.) To do this, see the procedure in *External Security Interface (RACROUTE) Macro Reference for MVS and VM*.

Controlling VTAM LU 6.2 Bind

With CICS 3.2, and VTAM 3.3, you can control which type 6.2 logical units can establish sessions with each other. This includes the ability to use RACF to specify the values used in password-on-bind processing. For more information, see *VTAM Programming for LU 6.2* and *APPC Management*.

To do this, take the following steps:

1. Ask your VTAM system programmer for the following information for each VTAM LU 6.2 pair:

   - The network ID and the LU identifiers for each member of the pair.
   - Optionally, the VTAM password and VTAM password interval for each LU 6.2 pair (RACF calls these the *session key* and *session interval*).

2. For each LU6.2 pair, create two profiles in the APPCLU class. On one system, do the following:

   ```
   RDEFINE  APPCLU  netid.luid1.luid2  UACC(NONE)
   ```

   On the other system, do the following:

   ```
   RDEFINE  APPCLU  netid.luid2.luid1  UACC(NONE)
   ```

   where:

   netid is the network ID or NETID, which is specified on the VTAM start option NETID (on MVS, this is in the ATCSTR*xx* member of SYS1.VTAMLST).

   luid1 and *luid2* are the LU names of the partners. In each case, the first LU name specified is the local LU name, and the second LU name is the partner LU name.

   For each profile created, the first LU name specified (*luid1*) is the *primary LU* on that system.

   Note: You should not specify an asterisk (*) or any other generic character for the first two qualifiers (*netid* and *luid*).

   When you create these profiles, specifying values for the SESSION segment provides much of the function of the profiles:

   - Session key, which is the VTAM password (SESSKEY suboperand on the SESSION operand).

Notes:

a. The session key can be required by the APPL statement.

b. If specified, the session key must be the same in both profiles of an LU 6.2 pair.

- The maximum number of days that the session key can exist before it must be changed (INTERVAL suboperand on the SESSION operand).
- Whether the LU pair is to be locked or unlocked (LOCK suboperand on the SESSION operand).

Note: To change any of the suboperands, issue the RALTER command with the SESSION operand specified.

3. Optionally, for maintenance purposes, give users and groups the access authority to the profiles:

```
PERMIT  profile-name  CLASS(APPCLU)
        ID(userid or group)  ACCESS(access-authority)
```

where *access-authority* is one of the following:

NONE Allows no access to the profile

READ Allows users to list the profile (for example, using the RLIST and SEARCH commands)

UPDATE Is the same as READ

CONTROL Is the same as READ

ALTER Allows users to change the profile (if the profile is discrete).

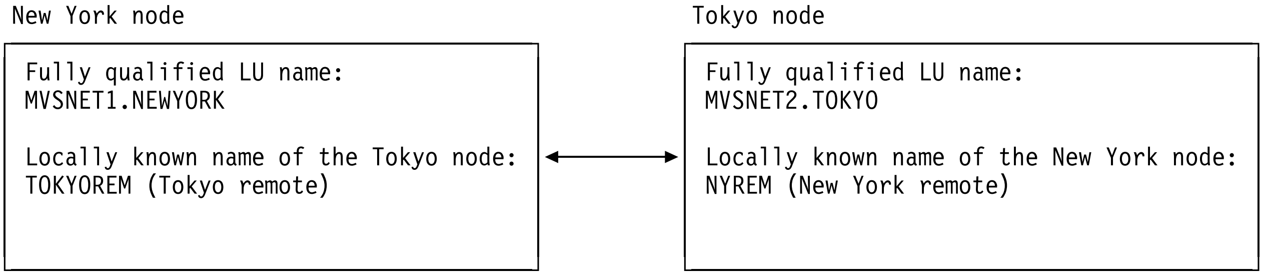
4. When you are ready to start using the protection defined in the profiles, activate the APPCLU class *on every system on which you want to use the APPCLU profiles:*

```
SETROPTS  CLASSACT(APPCLU)
```

Note: You cannot issue the SETROPTS RACLIST command for the APPCLU class. VTAM does this for you (using the RACLIST macro).

Example

Assume that there are two large nodes, one in New York, the other in Tokyo.



```
New York node                              Tokyo node

Fully qualified LU name:                   Fully qualified LU name:
MVSNET1.NEWYORK                            MVSNET2.TOKYO

Locally known name of the Tokyo node:      Locally known name of the New York node:
TOKYOREM (Tokyo remote)                    NYREM (New York remote)
```

On the New York node, take the following steps:

1. Define a profile for the Tokyo LU partner:

```
RDEFINE  APPCLU  MVSNET1.NEWYORK.TOKYOREM
                 SESSION(SESSKEY(KEY1))
                 UACC(NONE)
```

2. Activate the APPCLU class:

```
SETROPTS CLASSACT(APPCLU)
```

On the Tokyo node, take the following steps:

1. Define a profile for the New York LU partner:

```
RDEFINE  APPCLU  MVSNET2.TOKYO.NYREM
         SESSION(SESSKEY(KEY1))
         UACC(NONE)
```

2. Activate the APPCLU class:

```
SETROPTS CLASSACT(APPCLU)
```

Encryption of RACF User Passwords

By default, RACF stores RACF passwords on VM systems in the RACF database in a masked or hashed form. By either removing or modifying the ICHDEX01 exit routine, you can change RACF to use a software implementation of the data encryption standard (DES) algorithm to encrypt the passwords. You can also use this exit to replace the DES algorithm with any algorithm your installation desires (including the current masking algorithm).

RACF performs two different encoding functions:

- Password data encoding
- Password data comparison.

Encoding means that, given data in clear text and given an encryption key (which RACF constructs), the equivalent data is produced in encrypted form. RACF provides a “one-way” encoding. That is, data encrypted by RACF can only be decoded if the data is already known. See *RACF System Programmer's Guide* for additional details.

Comparison means that, given a password as entered by a user (in clear text form) and given a password as stored in the RACF database in encoded form, an indication as to whether they are equal or not is returned.

RACF performs password comparison in the following way:

- RACF encrypts the user-entered data using the DES algorithm and compares it against the stored version. If they are equal, RACF returns to the caller with an “equal” indication.

- RACF encodes the user-entered data using the current masking algorithm and compares it against the stored version. If they are equal, RACF returns to the caller with an “equal” indication.

By encoding the user-entered data against both the DES algorithm and the current masking algorithm, RACF allows the use of existing masked passwords until they can be replaced by the DES forms. (Note that eventually, all passwords will be in DES form because, whenever a user enters a new password, RACF uses the DES algorithm to encrypt it and store it. The ICHDEX01 exit routine can be used to change this processing.)

For compatibility with previous versions of RACF, a dummy ICHDEX01 exit routine is supplied with RACF. This dummy exit routine always returns a return code of 4, telling RACF to use only the old masking algorithm. To activate DES processing, the installation must delete the dummy exit routine from RACFLPA LOADLIB, and then restart RACF. Deleting the dummy exit routine should be done on all systems that share the RACF database after all those systems have all been converted to a version of RACF that supports the DES algorithm. Once all masked data has been converted to DES-encoded form, the installation can optionally install an ICHDEX01 exit routine that always returns a return code of 8, instructing RACF to ignore the old masking algorithm and use DES processing only.

For more information on changing or removing ICHDEX01, see *RACF System Programmer's Guide*.

Field-Level Access Checking

You can use RACF to control which users can access data in RACF profiles at the field level through *field-level access checking*. To do this, you create profiles in the FIELD class and permit users to the profiles.

Using field-level access checking, you can:

- 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.
- Allow all users to read or modify a particular field (or segment) of their *own* user profiles. To do this, specify ID(&RACUID) on the PERMIT command.

Note: RACF command processors and panels support field-level access checking only for fields in segments other than the base segments of RACF profiles, such as the DFP, DLFDATA, OVM, SESSION, and TSO segments. However, the ICHEINTY and RACROUTE REQUEST=EXTRACT macros can support field-level access checking for fields in any segment of any RACF profile. If your installation has written its own programs that use these macros to access the RACF database, you can modify these programs to implement field-level access checking.

To use field-level access checking, take the following steps:

1. Define profiles in the FIELD class:

   ```
   RDEFINE  FIELD  profile-name  UACC(NONE)
   ```

   where *profile-name* has the following format:

   ```
   profiletype.segmentname.fieldname
   ```

   where:

   profiletype is one of the following:

   - USER for user profiles
   - GROUP for group profiles
   - DATASET for data set profiles
   - Class name for general resource profiles.

segmentname is one of the following:

- TSO for TSO segments
- DFP for DFP segments
- DLFDATA for DLFDATA segments
- SESSION for SESSION segments
- CICS for CICS segments
- OPERPARM for OPERPARM segments
- LANGUAGE for LANGUAGE segments
- WORKATTR for WORKATTR segments
- OVM for OVM segments
- BASE for BASE segments (this is supported only by user-written code).

Notes:

a. This operand is also used on RACF commands to work with the segment.

b. See "Setting Up Field-Level Access for the OVM Segment" on page 247 for examples of field-level access checking for OVM segments.

fieldname is the name of the field to be protected as described in Table 22 on page 212.

For example, to control access to *all* fields in the TSO segment of all user profiles, issue the RDEFINE command and specify USER.TSO.* as the profile name. Before issuing this command, however, check the Special Note below:

```
RDEFINE  FIELD  USER.TSO.*  UACC(NONE)
```

When you specify a UACC of NONE, you prevent all users from accessing the TSO segment in all user profiles, including their own. Likewise, if you specify a UACC of READ, you allow all users to read the information contained in all fields of the TSO segment for all user profiles.

Special Note

Note that the profile name USER.TSO.* is a generic profile name. Before you issue the above command, generic profile checking for the FIELD class must be active. If it is not active, issue the SETROPTS GENERIC(FIELD) command before defining the generic profile.

To control access to specific fields in the TSO segment of user profiles, issue the RDEFINE command and specify the specific field as the third qualifier in the profile name. Use Table 22 on page 212 to determine which qualifier to use. For example, when changing the account number field in a TSO segment, users specify the ACCTNUM suboperand on the TSO operand of the ALTUSER command:

```
ALTUSER userid TSO(ACCTNUM(account-number))
```

According to Table 22 on page 212, to control access to the ACCTNUM suboperand, create a profile using the TACCNT qualifier:

```
RDEFINE  FIELD  USER.TSO.TACCNT  UACC(NONE)
```

2. Allow specific users or groups to have the appropriate access to the FIELD profile:

```
PERMIT USER.TSO.TLPROC CLASS(FIELD) ID(TSOADM)
       ACCESS(UPDATE)
```

The previous example shows how to create a profile that gives user ID TSOADM the authority to change the logon procedure (TLPROC field) in the profiles of all TSO users.

Note: You can also specify the value &RACUID with the ID operand on the PERMIT command for FIELD profiles. When you enter this value on the PERMIT command, you allow all users access to the specified field or segment of their own user profiles. For example, if you issue the following command, you allow all users to read the TLPROC field in the TSO segment of their own user profiles.

```
PERMIT USER.TSO.TLPROC CLASS(FIELD) ID(&RACUID)
       ACCESS(READ)
```

3. When you are ready to start using the protection defined in the profiles, activate the FIELD class:

   ```
   SETROPTS  CLASSACT(FIELD)
   ```

   Note: If you do not activate the FIELD class, only SPECIAL users can access fields in segments (other than the base segment) of RACF profiles.

4. It is recommended that you activate SETROPTS RACLIST processing for the FIELD general resource class. When you activate this function, you improve performance because I/O to the RACF database is reduced. For a complete description of this function, see “SETROPTS RACLIST Processing” on page 227.

   ```
   SETROPTS  RACLIST(FIELD)
   ```

   Note: If you activate SETROPTS RACLIST processing for the FIELD class, any time you make a change to a FIELD profile, you must also refresh SETROPTS RACLIST processing for the FIELD class for the change to take effect.

   ```
   SETROPTS  RACLIST(FIELD)  REFRESH
   ```

Table 22 (Page 1 of 2). Relationship of RACF Command Suboperands to FIELD Profile Names

To control the use of this suboperand:	Use this qualifier in FIELD profiles:
CICS Segment in User Profiles:	
OPCLASS	OPCLASS
OPIDENT	OPIDENT
OPPRTY	OPPRTY
TIMEOUT	TIMEOUT
XRFSOFF	XRFSOFF
DFP Segment in DATASET Profiles:	
RESOWNER	RESOWNER
DFP Segments in User and Group Profiles:	
DATAAPPL	DATAAPPL
DATACLAS	DATACLAS
MGMTCLAS	MGMTCLAS
STORCLAS	STORCLAS
DLFDATA Segment in General Resource Profiles (DLFCLASS Class):	
RETAIN	RETAIN
JOBNAMES	JOBNAMES
LANGUAGE Segment in User Profiles:	
PRIMARY	USERNL1
SECONDARY	USERNL2
OPERPARM Segment in User Profiles:	
ALTGRP	OPERALTG
AUTH	OPERAUTH
AUTO	OPERAUTO
CMDSYS	OPERCMDS
DOM	OPERDOM
KEY	OPERKEY
LEVEL	OPERLEVL
LOGCMDRESP	OPERLOGC
MFORM	OPERMFRM
MIGID	OPERMGID
MONITOR	OPERMON
MSCOPE	OPERMSCP
ROUTCODE	OPERROUT
STORAGE	OPERSTOR
UD	OPERUD
SESSION Segment in General Resource Profiles (APPCLU Class):	
CONVSEC	CONVSEC
INTERVAL	KEYINTVL
LOCK	SLSFLAGS
SESSKEY	SESSKEY
TSO Segment in User Profiles:	

Table 22 (Page 2 of 2). Relationship of RACF Command Suboperands to FIELD Profile Names

To control the use of this suboperand:	**Use this qualifier in FIELD profiles:**
ACCTNUM	TACCNT
DEST	TDEST
HOLDCLASS	THCLASS
JOBCLASS	TJCLASS
PROC	TLPROC
MAXSIZE	TMSIZE
MSGCLASS	TMCLASS
SECLABEL	TSOSLABL
SIZE	TLSIZE
SYSOUTCLASS	TSCLASS
UNIT	TUNIT
USERDATA	TUDATA
WORKATTR Segment in User Profiles:	
WANAME	WANAME
WABLDG	WABLDG
WADEPT	WADEPT
WAROOM	WAROOM
WAADDR1 through WAADDR4	WAADDR1 through WAADDR4
WAACCNT	WAACCNT
OVM Segment in User Profiles:	
UID	UID
HOME	HOME
PROGRAM	PROGRAM
FSROOT	FSROOT
OVM Segment in Group Profiles:	
GID	GID

Chapter 13. Selecting RACF Options on VM (SETROPTS)

This chapter describes the options that the RACF security administrator (who has the SPECIAL attribute) can specify on the SETROPTS command.

Note that you can also select SETROPTS options using the appropriate ISPF panels. For a complete description of the SETROPTS command, see *RACF Command Language Reference*.

Considerations for Selected SETROPTS Options

- Do not issue the SETROPTS TERMINAL(NONE) command unless you have RACF-protected enough terminals so that users can log on. SETROPTS TERMINAL(NONE) prevents users from logging on to unprotected terminals.
- Some classes have a default return code of 8. If such a class is activated, but no profiles are defined, user activity that requires access in that class is prevented.

  Do not activate a class with a default return code of 8, either explicitly (by name) or implicitly (by means of a shared POSIT value), unless you have defined profiles for that class.

  RACF prevents you from accidentally activating all classes by misusing the SETROPTS CLASSACT(*) operand.

  If security labels have been assigned to resource profiles, do not activate the SECLABEL class by using SETROPTS CLASSACT(SECLABEL) unless you have assigned appropriate security labels to appropriate users.

  To recover from such a situation, log on as a user with the system-SPECIAL attribute, specifying SYSHIGH as the current security label. Assign security labels or issue SETROPTS NOCLASSACT(SECLABEL).
- Do not issue the SETROPTS MLACTIVE(FAILURES) command unless you have assigned appropriate security labels to all users and to the resources that they must access.

  To recover from such a situation, log on as a user with the system-SPECIAL attribute, specifying SYSHIGH as the current security label. Assign security labels, or issue SETROPTS NOMLACTIVE.

Initial Setup

This section describes the SETROPTS options that you should consider doing when RACF is first installed.

Password Syntax Rules

If you have the SPECIAL attribute, you can establish up to eight password syntax rules to verify that new passwords meet the installation standards. These rules allow you to control:

- Minimum and maximum length of passwords
- Character content of installation-selected positions in the passwords

You establish these rules by using the RULE*n* suboperand specified by the PASSWORD operand of the SETROPTS command. The following example shows how you can establish a syntax rule for new passwords for your installation.

```
SETROPTS PASSWORD(RULE1(LENGTH(8) VOWEL(1,3,5:8) NUMERIC(2,4)))
```

The command establishes syntax rule RULE1. Syntax rule RULE1 specifies that new passwords must be 8 characters in length and contain vowels in positions 1, 3, 5, 6, 7, and 8 and numbers in positions 2 and 4. Thus, the password “A2E2EAEE” follows the rule, and “C3DMIER5” does not.

If you do not define a value for every position specified by the LENGTH value, the undefined positions can contain any combination of alphanumeric characters.

Note: If the RACF ISPF panels are installed, you should consider using the RACF ISPF panels to set up password syntax rules.

Maximum Password Change Interval

The INTERVAL suboperand specifies the system default for the number of days that a user's password is to remain valid. The following example specifies that each user's password remain valid for 60 days (as long as the system default for these users remains 60 days):

```
SETROPTS PASSWORD(INTERVAL(60))
```

This value becomes effective immediately as:

- A default value for new users whom you define to RACF through the ADDUSER command
- An upper limit for users who specify the INTERVAL operand on the PASSWORD command

When users are defined to RACF and have access to the system, they can use the INTERVAL operand of the PASSWORD command to set their own change interval to a value less than 30 or to a value less than that which you specified on the INTERVAL operand of the SETROPTS command (if you did so).

Note: When you invoke this option, the change interval values in existing user profiles are not modified. RACF uses the smaller of the two values.

The initial system default for the change interval is 30 days.

Extended Password and User ID Processing

If you have the SPECIAL attribute, you can specify the WARNING/NOWARNING, HISTORY/NOHISTORY, REVOKE/NOREVOKE, and INTERVAL options.

Use the PASSWORD option on the SETROPTS command to provide the following functions:

- The WARNING suboperand enables you to specify when RACF should issue a password expiration message. If you specify WARNING, RACF issues a message each time a user accesses the system a specified number of days before the password expires. The following example specifies that RACF issue a warning message 5 days before a password expires:

  ```
  SETROPTS PASSWORD(WARNING(5))
  ```

  If NOWARNING is in effect, RACF does not issue a warning message before a password expires.
- The HISTORY suboperand enables you to specify the number of previous passwords that RACF saves and compares with an intended new password. If there is a match with one of these previous passwords, or with the current password, RACF rejects the intended new password. If you increase the password HISTORY number, RACF saves and compares that number of passwords to the new password. If you reduce the password HISTORY number, passwords in the user profile that are beyond the newly specified HISTORY number are never deleted and continue to be used for comparison.

For example, if the HISTORY number is 12 and you reduce that HISTORY number to 8, RACF continues to compare the old passwords 9 through 12 with the intended new password.

The following example specifies that RACF save and compare 10 previous passwords, in addition to the current password, with an intended new password:

```
SETROPTS PASSWORD(HISTORY(10))
```

Note: When the user attempts to change his or her own password, RACF verifies that the intended new password does not match a previous entry in the history list, then saves the current password in the list, and changes the password to the new value. NOHISTORY specifies that the new password information is compared only to the current password. Any prior password history information is neither deleted nor changed.

The RACF password history list does **not** get updated with the password that was in effect when a user's password is reset.

- REVOKE enables you to specify how many consecutive password verification attempts RACF is to permit before it revokes a user ID on the next attempt. The following example specifies that RACF is to allow 4 consecutive invalid password attempts. A fifth invalid password attempt revokes the user ID:

  ```
  SETROPTS PASSWORD(REVOKE(4))
  ```

  After RACF revokes the user ID, you can activate the user ID with the RESUME operand of the ALTUSER command if you have the SPECIAL or group-SPECIAL attribute or are the owner of the profile. If NOREVOKE is in effect, consecutive invalid passwords are ignored.

Revoking Unused User IDs

The INACTIVE operand of the SETROPTS command causes RACF to revoke the user's right to use the system if the user ID has remained unused beyond a specified number of days. RACF revokes the user the next time the user attempts to enter the system.

The following example specifies that RACF revoke a user ID if it is unused for over 30 days:

```
SETROPTS INACTIVE(30)
```

Notes:

1. New users who never use the system are not revoked because of inactivity.
2. If a user has not logged on (or submitted a job) in 31 days, and you issue the SETROPTS INACTIVE(30) command, that user will be considered revoked. However, the user will not actually be revoked and the output of the LISTUSER command will not show that the user is revoked until the user next attempts to log on (or submit a job).
3. When you allow the user to once again start using the system (using the RESUME operand on the ALTUSER command), RACF resets the effective date with which the period of inactivity starts.

If NOINACTIVE is in effect, RACF does not check the user ID against an unused user ID interval.

If NOINITSTATS is in effect, the INACTIVE, REVOKE, HISTORY, and WARNING options cannot be used.

List-of-Groups Authority Checking

List-of-groups authority checking supplements the normal RACF access authority checking by allowing all groups of which a user ID is a member to enter into the access list checking process. This process replaces the checking that compares the current connect group with the resource's access list, and can expand a user's ability to access resources. If list-of-groups checking is active, then regardless of which group the user is logged on to, RACF recognizes the user's group-related authorities in other connect groups. If a user is in more than one group and tries to access a resource, RACF uses the highest authority allowed by the user's list of groups and the resource's access list.

Note: On VM, a user's current connect group is the default group specified in his or her user profile.

For example, the user is logged on to Group B (the current connect group) and tries to access a resource. The resource's access list does not contain the user's user ID or the group id for Group B, but it does contain the group id for Group A with an associated access authority of READ. If the user is a member of Group A (and Group B) and list-of-groups checking is active, the user can access the resource, even though the user is logged on to Group B. (This example assumes that other RACF checks, such as security classification checking, are met.)

Similarly, if list-of-groups checking is active, RACF recognizes the user's group-related attributes (such as group-SPECIAL) in other connect groups, regardless of which group the user is logged on to. However, the user still has each group-related attribute only within the “scope” of that group in which the user is assigned the attribute. (See Chapter 4, “Defining Groups” on page 79 for more information on the scope of the group.)

For example, in Figure 5 on page 65, say USER1 is also connected to GROUP3, but without group-SPECIAL for GROUP3. If list-of-groups checking is not active and USER1 logs on to GROUP3, RACF does not recognize that USER1 has group-SPECIAL authority to GROUP1 resources.

If list-of-groups checking is active and USER1 logs on to GROUP3, USER1 has group-SPECIAL authority to GROUP1 resources. However, USER1 does not have group-SPECIAL authority to GROUP3 resources. Likewise, if list-of-groups checking is active and USER1 logs on to GROUP1, USER1 has group-SPECIAL authority to GROUP1 resources, but not GROUP3 resources.

If you have the SPECIAL attribute, you can specify list-of-groups checking by using the GRPLIST option of the SETROPTS command as shown in the following example:

```
SETROPTS GRPLIST
```

To use current-connect-group checking, specify the NOGRPLIST option on the SETROPTS command.

We recommend the use of the GRPLIST option because it eases administration and minimizes the number of times the user might have to log off and log back on to access resources.

Considerations for OpenExtensions VM

OpenExtensions VM groups are RACF groups that have a GID defined in the OVM segment of the group's profile. Authority checks for access to OpenExtensions VM byte file system files and directories use the GID in the user's current connect group and up to NGROUPS_MAX supplementary groups (if SETROPTS GRPLIST is active) to make group access decisions. Authority checks for other VM resources use the RACF current group and list-of-groups support. See the ICHNGMAX macro in *RACF Macros and Interfaces* for more information on NGROUPS_MAX.

Setting the RVARY Passwords

If you have the SPECIAL attribute, you can specify passwords that the operator must use to respond to RVARY command requests to switch the RACF databases or change RACF status (activate/deactivate RACF). You can specify the passwords using the RVARYPW operand on the SETROPTS command. RACF allows you to specify separate passwords for switching the databases and for changing RACF status. The following example specifies **HAPPY** as the switch password and **RABBIT** as the status password:

```
SETROPTS RVARYPW(SWITCH(HAPPY) STATUS(RABBIT))
```

Password names must conform to the following rules:

- Passwords can be up to 8 characters in length
- Valid characters for names are alphabetic (A through Z), numeric (0 through 9), and national (#, @, and $).

When a RACF database is first initialized using IRRMIN00, the switch password and the status password are both set to YES.

Restricting the Creation of General Resource Profiles (GENERICOWNER Option)

If you have the SPECIAL attribute, you can restrict the creation of profiles in general resource classes. To do this:

Step 1 Issue a SETROPTS GENERICOWNER command.

Step 2 Define a ** profile for the class, with yourself as owner. (This prevents users lacking special authority from being able to define profiles in the class.)

Step 3 Define a *top* profile for each user, covering the subset of resources in the class which the user is allowed to create. Each user should be the owner of this *top* profile.

You have created an environment where the user can create only profiles that are *more specific* than the user's *top* profile. The only other users who can create profiles in the user's subset of the class are:

- A user with SPECIAL authority
- A user who has group-SPECIAL authority over a user who owns the *top* profile.

For example, assume that neither JOE nor RHONDA have the SPECIAL or group-SPECIAL attribute. If the GENERICOWNER option is in effect, and user RHONDA is the owner of a JESSPOOL profile called NODEA.RHONDA.**, JOE cannot create profile NODEA.RHONDA.DATA.**, even though JOE has the CLAUTH(JESSPOOL) attribute.

Note: The GENERICOWNER operand does not affect the DATASET class. It cannot be activated for individual classes. When active, GENERICOWNER affects all general resource classes except the PROGRAM class and general resource grouping classes.

For example, when working with general resource grouping classes, assume that profile A* exists in the TERMINAL class and is owned by a group that user ELAINE does not have group-SPECIAL authority to. If the GENERICOWNER option is in effect, it will prevent user ELAINE from defining a more specific profile in the member class (for example, by using the command `RDEF TERMINAL AA*`). However, having the GENERICOWNER option in effect will **not** prevent user ELAINE from defining a profile if specified on the ADDMEM operand for the grouping class profile (such as with the command `RDEF GTERMINL` *`profile-name`* `ADDMEM(AA*)`).

You can alternatively choose to make a group the owner of the *top* profile for a given subset in the class. In this case, only a user with group-SPECIAL authority for the group, or with SPECIAL authority, can create profiles in the subset.

The *top* profile must end in * or **. More specific profiles are profiles that match the less specific *top* profile name character for character, up to the ending * or ** in the less specific name.

In a search for the less specific profile a match is found if *both:* of the following are true:

- The profile name ends in * or **
- All characters preceding the * or ** exactly match the corresponding characters in the resource name.

For example, to allow BOB to RDEFINE A.B in the JESSPOOL class, you need profile A.* in the JESSPOOL class, which is owned by BOB.

To cancel this option, specify NOGENERICOWNER on the SETROPTS command.

Attention

Issuing SETROPTS GENERICOWNER can prevent users with the CLAUTH attribute in general resource classes from creating profiles as they are accustomed to. Therefore, make these users OWNER of appropriate “top” generic profiles in the class. For an example, see “Delegating Authority to Profiles in the FACILITY Class” on page 205.

Activating and Deactivating General Resource Classes

If you have the SPECIAL attribute, you can specify that RACF provides access authorization checking for general resource classes. You can specify this option for selected general resource classes with the CLASSACT operand of the SETROPTS command. The following example shows how to specify RACF access authorization checking for the VMMDISK and VMRDR resource classes.

```
SETROPTS CLASSACT(VMMDISK VMRDR)
```

It is not recommended that you activate all RACF classes. You should only activate the classes that are important to your installation, as some classes have a default return code of 8. Those classes should only be activated after you have defined the necessary profiles to allow access to resources. When using the SETROPTS CLASSACT(*) operand, RACF prevents you from accidentally activating classes that have a default return code of 8.

For information on activating protection for specific general resource classes, check the index of this book for the class name.

If you have the SPECIAL attribute, you can also specify the NOCLASSACT operand on the SETROPTS command. This operand indicates that RACF performs no access authorization checking for selected general resource classes. If you specify NOCLASSACT(*), RACF does not perform access authorization checking for any of the classes in the class descriptor table. However, you can still define resource profiles to RACF using the ADDDIR, ADDFILE, and RDEFINE commands.

Special Considerations for OpenExtensions VM

The following classes are defined only for auditing OpenExtensions VM security events and are not used for authorization checking:

DIRACC
DIRSRCH
FSOBJ
FSSEC
PROCESS

No profiles can be defined in these classes. They are used to define the auditing options for OpenExtensions VM security events. The classes do not need to be active to control auditing.

SETROPTS LOGOPTIONS can be used to specify logging options for all of the classes. Additionally, SETROPTS AUDIT options are used to control auditing of some events for the FSOBJ class. For more information about the SETROPTS command, see *RACF Command Language Reference.*

Generic Profile Checking and Generic Command Processing

If you have the SPECIAL attribute, you can activate or deactivate generic profile checking either on a class-by-class basis or for all classes. You can specify this option with the GENERIC and NOGENERIC operands of the SETROPTS command. The following example shows how to activate generic profile checking for the VMCMD class.

```
SETROPTS GENERIC(VMCMD)
```

If you specify GENERIC(*), you activate generic profile checking for the VMCMD class plus all classes in the class descriptor table except resource group classes (such as GTERMINL).

If you want to perform maintenance on the generic profiles in the RACF database, you may want to temporarily deactivate generic profile checking but allow RACF command processors to update generic profiles. You can specify this environment with the NOGENERIC and GENCMD operands of the SETROPTS command. The following example shows how to specify this environment for the VMCMD class.

```
SETROPTS NOGENERIC(VMCMD) GENCMD(VMCMD)
```

NOGENERIC and NOGENCMD are in effect when a RACF database is first initialized using IRRMIN00.

Note: On VM, generic profiles are recommended only if you also specify SETROPTS GENLIST for the class.

Using SETROPTS STATISTICS for Statistics Collection

Using the SETROPTS STATISTICS option does the following:

- RACF maintains two sets of statistics in a discrete resource profile. One set counts all activity for the resource or profile; the other set counts activity for each entry in the access list. It can be difficult to compare the two sets of statistics meaningfully, unless you understand how RACF maintains the statistics. See the “STATISTICS Example” for clarification.
- If a specific resource has unique security concerns, you should protect it with a discrete profile.

  To see how that resource is being accessed and how many times it is being accessed, you can initiate STATISTICS. Remember that the initiation of STATISTICS is *system-wide* for all discrete profiles within a particular resource class across your system. Depending on the number of discrete profiles in the various resource classes, turning on STATISTICS may negatively affect performance.

STATISTICS Example

To help you understand how RACF maintains statistics, consider the following:

- USER1.DATA is a data set profile.
- USER1.DATA has a universal access (UACC) of READ.
- USER2 is in the access list with READ authority.
- USER3 is in the access list with UPDATE authority.
- GROUP1 is in the access list with READ authority.
- GROUP2 is in the access list with UPDATE authority.
- USER4 belongs to both groups, GROUP1 and GROUP2.
- There is no entry for &RACUID.* in the global access table.

If USER1 reads USER1.DATA, the overall READ count in the profile increases by one. No counts in the access list are changed, because access lists are not used when users process their own data.

If USER2 reads the data set, two counts are updated: the overall READ count, and the count in USER2's access list entry.

If USER3 reads the data set, two counts are updated: the overall READ count, and the count in USER3's access list entry (even though the entry says UPDATE). The counts in the access list merely record that access was granted by that entry. The access granted can be as specified by the entry, or a lower level, as in this example.

If list-of-groups processing is active (through SETROPTS GRPLIST) and USER4 reads the data set, RACF examines the access list to see if any of USER4's groups are in the list. If any of the groups is found, the entry with the highest authority is used. In this case, the access list entry for GROUP2 (UPDATE) increases, along with the overall READ count for the profile.

If any other user or group reads the data set, it gains access because of the universal access of READ, and the overall READ count increases. If any user with OPERATIONS authority updates the data set, the overall UPDATE count increases.

Using Options in RACINIT Statistics Collection

A user with a SPECIAL attribute can instruct RACF to record statistics during the RACINIT processing; RACINIT instructions are issued when a user is logging on to a system or a batch job is entering a system. The statistics RACF maintains are:

- The date and time RACINIT is issued for a particular user
- 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.

The statistics must be maintained if you intend to use the INACTIVE, REVOKE, HISTORY, and WARNING options.

If, for your system, you do not need all the statistics, you do not need to use the above four options, and you have the SPECIAL attribute, you may issue SETROPTS NOINITSTATS which will reduce the RACF database I/O associated with RACINIT function.

If NOINITSTATS is in effect, the INACTIVE, REVOKE, HISTORY, and WARNING options cannot be used.

If you have the SPECIAL attribute, you can also specify the INITSTATS operand on the SETROPTS command to indicate that you want RACF to record RACINIT statistics as shown in the following example:

```
SETROPTS INITSTATS
```

INITSTATS is in effect when a RACF database is first initialized using IRRMIN00.

Bypassing Resource Statistics Collection

If you have the SPECIAL attribute, you can request that RACF bypass the recording of statistical information in discrete profiles for the DATASET class on MVS and classes defined in the class descriptor table on both MVS and VM. You specify this option with the NOSTATISTICS operand of the SETROPTS command.

The statistics you can bypass include:

- Date the resource was last referenced
- Date the resource was last updated (not recorded for terminals)
- Number of times the resource was accessed for each of the following access authorities: ALTER, CONTROL, UPDATE, and READ (only READ count is recorded for terminals)
- Number of times each user or group in the access list has accessed the resource.

If you have the SPECIAL attribute, you can also specify the STATISTICS operand on the SETROPTS command and identify the class(es) for which you want RACF to record statistical information.

If you specify an asterisk (*), you activate the recording of statistical information for all resource classes.

When a RACF database is first initialized using IRRMIN00, STATISTICS is in effect for the DATASET, DASDVOL, TAPEVOL, and TERMINAL classes. Because statistics recording has an impact on system performance, it is recommended that you deactivate this option until your installation evaluates the need to use it versus the potential performance impact. For more information, see *RACF System Programmer's Guide*.

Global Access Checking

If you have the SPECIAL attribute, you can activate or deactivate global access checking on a class-by-class basis or for all classes. You can specify this option with the GLOBAL and NOGLOBAL operands of the SETROPTS command. The following example shows how to activate global access checking for the VMMDISK class.

```
SETROPTS GLOBAL(VMMDISK)
```

If you specify GLOBAL(*), you activate global access checking for the DATASET class plus all classes in the class descriptor table except group resource classes.

When you use the SETROPTS command to activate (or reactivate) global access checking for a class, RACF builds (or updates) the in-storage global access checking tables. However, you can use the RDEFINE and RALTER commands to maintain the global access checking tables, whether or not the global access checking option is active for a class.

NOGLOBAL is in effect when a RACF database is first initialized using IRRMIN00.

Note: Global access checking will be ignored for any class that has been brought into storage by a SETROPTS RACLIST command.

Daily Administration

This section describes the SETROPTS options that are used during daily administration.

Activating Shared In-Storage Profiles for General Resources

RACF provides SETROPTS GENLIST processing and SETROPTS RACLIST processing to allow your installation to reduce the amount of storage used by general resource profiles and the amount of processing associated with retrieving profiles from the RACF database. The following sections describe these functions.

Notes:

1. RACF does not allow you to specify SETROPTS GENLIST and SETROPTS RACLIST for the same general resource class.
2. If you RACLIST a class, global access checking is disabled for the class, because there is no performance benefit to checking the global access table compared with an in-storage profile.
3. If RACF 1.9 or later is installed, whenever you re-IPL the system, RACF automatically reactivates SETROPTS GENLIST and RACLIST processing for the classes for which this was previously requested.

SETROPTS RACLIST Processing on Shared Systems

The RACLIST processing for SETROPTS applies only to the system (VM or MVS) on which you issue the SETROPTS command. If your installation has two or more systems sharing a RACF database, you must issue the SETROPTS command on all systems to have the RACLIST done on all systems. In a multiple RACF service machine environment, you must also issue the SETROPTS command to each service machine that shares the RACF database. (To coordinate the command across multiple RACF service machines, refer to the RAC command information in *RACF Command Language Reference*.)

However, if you do not issue the SETROPTS command with the RACLIST option on a system sharing a RACF database and that system needs to re-IPL, RACLIST is performed for that system when a re-IPL occurs.

SETROPTS GENLIST Processing

If you have the SPECIAL attribute, you can activate SETROPTS GENLIST processing. Activate this function for general resource classes that contain a small number of frequently referenced generic profiles. When you activate SETROPTS GENLIST processing, you enable the sharing of in-storage generic profiles for the classes you specify. For a list of the classes eligible for GENLIST processing, see the description of the CDT in *RACF Macros and Interfaces*.

To activate this function, issue the SETROPTS command with the GENLIST operand. The following example shows how to activate SETROPTS GENLIST processing for the TERMINAL class.

```
SETROPTS  GENLIST(TERMINAL)
```

After you activate SETROPTS GENLIST processing for a general resource class, RACF copies a generic profile in that class from the RACF database into common storage the first time an authorized user requests access to a resource protected by it. The profile is then retained in common storage and is available to all authorized users, thereby saving real storage because the need to retain multiple copies of the same profile (one copy for each requesting user) in common storage is eliminated. Also, because RACF does not have to retrieve the profile each time a user requires access to it, this function saves processing overhead.

Note that a general resource class must be active before you can activate SETROPTS GENLIST processing for that class. If the class is not active, issue the SETROPTS command with both the GENLIST and CLASSACT operands and specify the desired class. The following example shows how to activate the TERMINAL class and SETROPTS GENLIST processing for that class on the same command.

```
SETROPTS  CLASSACT(TERMINAL)  GENLIST(TERMINAL)
```

If you have the SPECIAL attribute, you can also deactivate SETROPTS GENLIST processing for general resource classes. To deactivate this function, issue the SETROPTS command with the NOGENLIST operand and the selected general resource class(es).

NOGENLIST is the default and is in effect for all eligible classes defined in the CDT when a RACF database is first initialized using IRRMIN00.

For more information about SETROPTS GENLIST processing, see *RACF System Programmer's Guide*.

SETROPTS GENLIST Processing on Shared Systems: If your installation has two or more VM or MVS systems sharing a RACF database, you only need to issue the SETROPTS GENLIST command on one of those systems. SETROPTS GENLIST processing will be automatically propagated to all systems sharing the database and to every service machine in a multiple RACF service machine environment.

Refreshing Profiles for SETROPTS GENLIST Processing: If your installation has activated SETROPTS GENLIST processing for a particular resource class, you will need to refresh in-storage profiles for this processing when you make changes to one of these profiles. Refreshing profiles for SETROPTS GENLIST processing ensures that the most current copy of a profile resides in common storage and is available for RACF authorization checking. To refresh profiles for this processing, issue the SETROPTS command with the GENERIC and REFRESH operands and specify the appropriate resource class(es). For more information, see “Refreshing In-Storage Generic Profile Lists” on page 230.

SETROPTS REFRESH Processing on Shared Systems

The refresh operation for SETROPTS processing applies only to the system (VM or MVS) on which you issue the SETROPTS command. If your installation has two or more systems sharing a RACF database, you must issue the SETROPTS command on all systems to have the refresh done on all systems. In a multiple RACF service machine environment, you must also issue the SETROPTS command to each service machine that shares the RACF database. (To coordinate the command across multiple RACF service machines, refer to the RAC command information in *RACF Command Language Reference*.)

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

SETROPTS RACLIST Processing

If you have the SPECIAL attribute, you can activate SETROPTS RACLIST processing. Activate this function when a general resource class contains a small number of frequently referenced profiles for which you cannot use global access checking. When you activate SETROPTS RACLIST processing, you enable the sharing of both in-storage discrete and in-storage generic profiles for the classes you specify. For a list of the classes eligible for RACLIST processing, see the description of the CDT in *RACF Macros and Interfaces*.

To activate this function, issue SETROPTS with the RACLIST operand. The following example shows how to activate SETROPTS RACLIST processing for the TERMINAL class.

```
SETROPTS  RACLIST(TERMINAL)
```

When you activate SETROPTS RACLIST processing for a general resource class, RACF copies both discrete and generic profiles for the specified class into common storage. These profiles are available to all authorized users, thereby eliminating the need for RACF to retrieve a profile each time a user requests access to a resource protected by that profile. As a result, when you activate this function, you reduce processing overhead.

Note that a general resource class must be active before you can activate SETROPTS RACLIST processing for that class. If the class is not active, issue the SETROPTS command with both the RACLIST and CLASSACT operands and specify the desired class. The following example shows how to activate the TERMINAL class and SETROPTS RACLIST processing for that class on the same command.

```
SETROPTS  CLASSACT(TERMINAL)  RACLIST(TERMINAL)
```

Note: If you RACLIST a class, global access checking is disabled for the class, because there is no performance benefit to checking the global access table compared with an in-storage profile.

If you have the SPECIAL attribute, you can also deactivate SETROPTS RACLIST processing for general resource classes. To deactivate this option, issue SETROPTS with the NORACLIST operand and the selected general resource class(es).

NORACLIST is the default and is in effect for all eligible classes defined in the CDT when a RACF database is first initialized using IRRMIN00.

For more information about SETROPTS RACLIST processing, see *RACF System Programmer's Guide*.

Notes:

1. SETROPTS RACLIST processing does not affect the use of the RACROUTE REQUEST=LIST macro. An installation can continue to code the RACROUTE REQUEST=LIST macro to use RACROUTE REQUEST=FASTAUTH for fast-path authorization checking. However, because the SETROPTS RACLIST command (unlike the RACROUTE REQUEST=LIST macro) uses only RACROUTE REQUEST=FASTAUTH to perform authorization checking, an installation cannot use this command to invoke the RACROUTE REQUEST=FASTAUTH facility.
2. If your installation uses the RACROUTE REQUEST=LIST macro to load profiles for a general resource class into users' address spaces and also uses SETROPTS RACLIST to load profiles for the same resource class into common storage, RACF will use the profiles in a user's address space to perform authorization checking and the result is a waste of common storage. Therefore, your installation should not issue SETROPTS RACLIST for a general resource class if the RACROUTE REQUEST=LIST macro was already used to load profiles into storage for that class.

SETROPTS RACLIST Processing on Shared Systems

The RACLIST processing for SETROPTS applies only to the system (VM or MVS) on which you issue the SETROPTS command. If your installation has two or more systems sharing a RACF database, you must issue the SETROPTS command on all systems to have the RACLIST done on all systems. In a multiple RACF service machine environment, you must also issue the SETROPTS command to each service machine that shares the RACF database. (To coordinate the command across multiple RACF service machines, refer to the RAC command information in *RACF Command Language Reference*.)

However, if you do not issue the SETROPTS command with the RACLIST option on a system sharing a RACF database and that system needs to re-IPL, RACLIST is performed for that system when a re-IPL occurs.

Refreshing Profiles for SETROPTS RACLIST Processing: Any changes made to discrete or generic profiles activated for SETROPTS RACLIST processing become effective only when you issue the SETROPTS command with both the RACLIST and REFRESH operands. You can refresh these profiles if you have one of the following:

- The SPECIAL attribute
- CLAUTH authority to the general resource class you specify on the RACLIST operand.

To activate this option, issue SETROPTS with the RACLIST and REFRESH operands and specify the general resource class that contains the profiles you want to refresh. Note that you must issue this command each time you want RACF to perform the refresh process. The following example shows how to activate refreshing of SETROPTS RACLIST processing for the TERMINAL classes.

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

If you specify NORACLIST together with a general resource class on the SETROPTS command, you remove the profiles from storage for the class(es) that you specify.

SETROPTS REFRESH Processing on Shared Systems

The refresh operation for SETROPTS processing applies only to the system (VM or MVS) on which you issue the SETROPTS command. If your installation has two or more systems sharing a RACF database, you must issue the SETROPTS command on all systems to have the refresh done on all systems. In a multiple RACF service machine environment, you must also issue the SETROPTS command to each service machine that shares the RACF database. (To coordinate the command across multiple RACF service machines, refer to the RAC command information in *RACF Command Language Reference*.)

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

Refreshing In-Storage Generic Profile Lists

If you have the SPECIAL, AUDITOR, or OPERATIONS attribute, or if you have CLAUTH authority for the classes specified, you can initiate the refreshing of in-storage generic profile lists by specifying the GENERIC and REFRESH operands on the SETROPTS command.

When you specify GENERIC and REFRESH, you also specify one or more classes for which you want RACF to refresh in-storage generic profile lists. This causes all the in-storage generic profiles within the specified general resource class (except those in the global access checking table) to be replaced with new copies from the RACF database. Note that you must issue this command each time you want RACF to perform the refresh process.

The following example shows how to refresh in-storage generic profiles for the VMCMD and TERMINAL classes.

```
SETROPTS GENERIC(VMCMD TERMINAL) REFRESH
```

If you use SETROPTS GENLIST to activate shared in-storage generic profiles for a general resource class, RACF refreshes the profiles as well as the profile lists for that class when you specify the class with GENERIC and REFRESH. See “Activating Shared In-Storage Profiles for General Resources” on page 225 for more information.

If you specify GENERIC(*), RACF refreshes profile lists for the DATASET class and all active classes in the class descriptor table except group resource classes.

When you initiate the refresh procedure, the refresh may not happen immediately. RACF refreshes a generic profile list the next time that the list is needed for an authorization check.

Note: The VM system does not use either the RACROUTE REQUEST=LIST or the RACROUTE REQUEST=FASTAUTH macro. However, a VM user can write an exit routine that invokes them.

If you specify NOGENERIC on the SETROPTS command, RACF stops using in-storage generic profile lists but does not immediately delete them. On VM, RACF deletes the profile lists only when you again specify GENERIC. When you specify GENERIC, RACF rebuilds the profile lists.

Note: You must have the SPECIAL attribute to specify the GENERIC option by itself. However, when you issue the REFRESH operand with the GENERIC operand (to effect a refresh without altering the status of the system), you do not require the SPECIAL attribute. However, you must have the group-SPECIAL, group-AUDITOR, group-OPERATIONS, AUDITOR, or OPERATIONS attribute, or you must have CLAUTH authority for the classes specified.

SETROPTS REFRESH Processing on Shared Systems

The refresh operation for SETROPTS processing applies only to the system (VM or MVS) on which you issue the SETROPTS command. If your installation has two or more systems sharing a RACF database, you must issue the SETROPTS command on all systems to have the refresh done on all systems. In a multiple RACF service machine environment, you must also issue the SETROPTS command to each service machine that shares the RACF database. (To coordinate the command across multiple RACF service machines, refer to the RAC command information in *RACF Command Language Reference*.)

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

Refreshing Global Access Checking Lists

If you have the SPECIAL attribute, you can initiate the refreshing of global access checking lists by specifying the GLOBAL and REFRESH operands on the SETROPTS command. When you specify GLOBAL and REFRESH, also specify the class for which you want RACF to refresh global access checking lists. Note that you must issue this command each time you want RACF to perform the refresh process. The following example specifies that you want RACF to refresh global access checking lists for the VMCMD and TERMINAL classes.

```
SETROPTS GLOBAL(VMCMD TERMINAL) REFRESH
```

If you specify GLOBAL(*), RACF refreshes access checking lists for the DATASET class and all active classes in the class descriptor table.

If you specify NOGLOBAL, you disable global access checking for the class that you specify.

SETROPTS REFRESH Processing on Shared Systems

The refresh operation for SETROPTS processing applies only to the system (VM or MVS) on which you issue the SETROPTS command. If your installation has two or more systems sharing a RACF database, you must issue the SETROPTS command on all systems to have the refresh done on all systems. In a multiple RACF service machine environment, you must also issue the SETROPTS command to each service machine that shares the RACF database. (To coordinate the command across multiple RACF service machines, refer to the RAC command information in *RACF Command Language Reference*.)

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

Special Purpose Options

This section describes the SETROPTS options that are useful for special purposes. You can do them at any time, but you might not wish to do them when RACF is first installed.

Universal Access Authority for Undefined Terminals

If you have the SPECIAL attribute, you can specify the universal access authority that RACF uses when users attempt to log on to VM from terminals that are not defined to RACF. You can specify this option with the TERMINAL operand of the SETROPTS command as shown in the following example:

```
SETROPTS TERMINAL(READ|NONE)
```

Attention

Before you specify NONE, be sure that you define your terminals to RACF and give the appropriate users and groups proper authorization to use them. Otherwise, users will not be able to access the terminals.

When you specify READ or NONE, you establish a default universal access authority for all users to undefined terminals on your system. If you specify READ, all users can access all terminals on your system (if allowed by the security classifications of the terminals). If you specify NONE, only users and groups that you authorize to use a terminal through its access list can use it. If you do not specify either READ or NONE, the default value is READ. For more detailed information, see “Protecting Terminals on VM” on page 180.

For undefined terminals, READ access authority is in effect when a RACF database is first initialized using IRRMIN00.

Activating Security Classification of Users and Data

If you have the SPECIAL attribute, you can activate security classification of users and data by specifying CLASSACT(SECDATA) or CLASSACT(SECLABEL) on the SETROPTS command as shown in the following examples:

```
SETROPTS CLASSACT(SECDATA)
SETROPTS CLASSACT(SECLABEL)
```

Security classification of users and data allows an installation to impose additional access controls on resources by defining security levels, security categories, and security labels for both users and resources.

Note: You can create profiles in the SECDATA and SECLABEL classes, and specify security classifications in resource and user profiles, without actually activating the SECDATA and SECLABEL classes. When authorization checking occurs, the security classifications will be ignored by RACF. This allows you to “label” the profiles without enforcing the security classifications.

If you have the SPECIAL attribute, you can also deactivate security classification of users and data by specifying NOCLASSACT(SECDATA) or NOCLASSACT(SECLABEL), as appropriate, on the SETROPTS command.

NOCLASSACT(SECDATA) and NOCLASSACT(SECLABEL) are in effect when a RACF database is first initialized using IRRMIN00.

Establishing a System-Wide Maximum VTAM Session Interval

If you have the SPECIAL attribute, you can set the maximum number of days that any session segment password in an APPCLU profile can go without being changed. If any APPCLU profile has a higher interval limit, the SETROPTS value is used instead.

To set the limit, enter the following command:

```
SETROPTS SESSIONINTERVAL(n)
```

where *n* is the maximum number of days that can be specified and must be between 1 and 32767.

To remove the system-wide maximum (allowing any value in the profiles to take effect), enter the following command:

```
SETROPTS NOSESSIONINTERVAL
```

Temporarily Preventing Significant RACF Activity

If you have the SPECIAL attribute, you can prevent users other than SPECIAL users, console operators, and started procedures from logging on, starting new jobs, or accessing resources. (This prevents the issuing of the RACINIT, RACHECK, or RACDEF macros, or an equivalent RACROUTE macro.)

To do this, enter the following command:

```
SETROPTS MLQUIET
```

To cancel this option, specify NOMLQUIET on the SETROPTS command.

Attention

Do not specify SETROPTS MLQUIET if any system using the RACF database is not at the necessary software level for B1 support. See Page 11.

Use of the MLQUIET option may prevent a successful IPL of these systems. In addition, if no systems using the RACF database are capable of B1 support, you may have to IPL with RACF inactive and update the database with BLKUPD in order to turn off the MLQUIET option.

Options Related to Security Labels

This section describes the SETROPTS options that require that the SECLABEL class be active.

Restricting Changes to Security Labels

If you have the SPECIAL attribute, you can prevent users who do not have the SPECIAL attribute from doing either of the following:

- Specifying or changing a security label in a resource profile
- Changing the profiles named SECLEVEL or CATEGORY, or changing any profile in the SECLABEL class, such that the definition of a security label changes.

To place this control into effect, issue the following command:

```
SETROPTS SECLABELCONTROL
```

When the SECLABELCONTROL option is in effect, only certain users can specify the SECLABEL operand on RACF commands:

- Users with the system-SPECIAL attribute can specify the SECLABEL operand on any RACF command.
- Users with the group-SPECIAL attribute can specify the SECLABEL operand only on the ADDUSER and ALTUSER commands when adding a user to a group within their scope of control. Also, group-SPECIAL users must be permitted to the SECLABEL profiles with at least READ access authority.
- Users without the SPECIAL attribute cannot specify the SECLABEL operand.

To cancel this option, specify NOSECLABELCONTROL on the SETROPTS command.

Preventing Users from Copying Data from One Security Label to a Lower Security Label

If you have the SPECIAL attribute, and if the SECLABEL class is active, you can prevent users from copying data from a resource with one security label to a resource with a lower security label. Users will still be able to copy data from a lower security label to the security label the user is currently logged on with.

To do this, enter the following command:

```
SETROPTS MLS(FAILURES)
```

You can also specify MLS(WARNING), which allows the user request, but sends a warning message to the user and the security administrator.

To cancel the MLS option, specify NOMLS on the SETROPTS command.

Note: Do not specify SETROPTS MLS if any system using the RACF database is not at the necessary software level for B1 support. Use of the MLS option should not cause problems on these systems, but it will not provide full protection on these systems. See Page 11.

Compatibility Mode for Security Labels

If you are using SECLABELs on your system, and you observe SECLABEL failures for some calls at the designated security console or in audit records, the reason may be that the call was not made by a 1.9 or later caller, or the caller was not using or was unable to use the 1.9 or later protocols.

To investigate the source of the failure, obtain a copy of the RACF Report audit record. Examine the EVENT and QUALIFIER fields for the call to see if the failure occurred because of insufficient security-label authority. Next examine the “token status =” field to the right of the QUALIFIER field. If the field is identified as being “created by a pre-1.9 call,” this means the ACEE (an area created to identify the call's security environment) was created by a pre-1.9 caller, or by a caller that was either not using or unable to use the 1.9 or later protocols. As a result, RACF failed the SECURITY label authorization check.

If this was the case, and you want to have SECLABEL authorization checks succeed for those callers who are not using 1.9 or later protocols, you may be able to use the COMPATMODE keyword on the SETROPTS command to do so. Specifying COMPATMODE allows the caller to access the resources it needs, providing the user is in the access list of a SECLABEL (it need not be one that protects the resource) that is higher than or equal to the SECLABEL that protects the resource.

To establish COMPATMODE, enter the following command:

```
SETROPTS COMPATMODE
```

NOCOMPATMODE is in effect when a RACF database is first initialized using IRRMIN00.

Attention

If you put COMPATMODE into effect, it affects all pre-1.9 callers.

Enforcing Multilevel Security

If you have the SPECIAL attribute, and if the SECLABEL class is active, you can fully enforce multilevel security. To check your system configuration, see Page 11. Multilevel security requires the following:

- All work entering the system must be run by a RACF-defined user.
- A security label must be assigned to all work entering the system, including batch jobs and users logging on to VM and to any application that supports security labels when users log on.
- A security label must be assigned to all profiles in the following classes:

  DIRECTRY
  FILE
  TAPEVOL
  TERMINAL
  USER
  VMMDISK
  VMSEGMT
  WRITER

  To do this, enter the following command:

  ```
  SETROPTS MLACTIVE
  ```

  You can also specify MLACTIVE(WARNING), which allows the users to log on or submit jobs. MLACTIVE(WARNING) sends a warning message to the user and to the security administrator when the user attempts to:

  - Enter the system without a security label
  - Access a resource in one of the forementioned classes and the resource has not been assigned a security label.

To cancel the MLACTIVE option, specify NOMLACTIVE on the SETROPTS command.

Note: Do not specify SETROPTS MLACTIVE if any system using the RACF database is not at the necessary software level for B1 support. See Page 11.

Use of the MLACTIVE option should not cause problems on these systems if you assign a default security label to all users and to all resources in the classes listed above. However, since users on these systems cannot supply a security label when they log on or run a batch job, the usefulness of requiring that everything be labelled is reduced. Also, some error recovery scenarios may require the security administrator to log on specifically at the SYSHIGH level, and this will not be possible unless at least one system sharing the database has the B1-support software.

Attention

Do not issue the SETROPTS MLACTIVE(FAILURES) command unless you have assigned appropriate security labels to users and to the resources that they must access. To recover from such a situation, logon as a user with the system-SPECIAL attribute, specifying SYSHIGH as the current security label. Then either assign security labels, or issue SETROPTS NOMLACTIVE.

Preventing Changes to Security Labels

If you have the SPECIAL attribute, you can prevent users from changing the classification of data while the data is in use. Specifically, you can do all of the following:

- Prevent any user from changing the security label of a RACF profile.
- Prevent any user from changing a SECLABEL profile such that the definition of the security label changes. (Users cannot change the security level or security categories associated with the security label.) Other changes to the SECLABEL profile, such as changes to the access list, are still allowed.
- On VM, prevent any user from changing the security label of a spool file with the CHANGE command.

To do this, enter the following command:

```
SETROPTS MLSTABLE
```

To cancel this option, specify NOMLSTABLE on the SETROPTS command.

Note: If you must change security labels while the system is in multilevel stable state, you can issue SETROPTS MLQUIET before making the changes. See “Temporarily Preventing Significant RACF Activity” on page 233.

Chapter 14. Controlling OpenExtensions z/VM Security

z/VM with RACF provides security at the Portable Operating System Interface (POSIX) 1003.1 level. This support also includes the POSIX 1003.1a draft 6 functions needed for a Department of Defense (DoD) C2 security level.

POSIX, a standard introduced by the Institute of Electrical and Electronics Engineers (IEEE), defines a standard set of interfaces through which a program can request services from the operating system it is running on. A *POSIX-compliant system* adheres to the specifications documented for each function defined, regardless of the differences in underlying implementations. When application developers design programs that use only the system interfaces defined by POSIX, these programs can be ported easily to other operating systems and programs that are POSIX-compliant. OpenExtensions is z/VM's implementation of POSIX.

OpenExtensions security information is stored and maintained in the RACF database in OVM segments of USER and GROUP profiles. You can work with the information in the OVM segment using the ADDUSER, ALTUSER, LISTUSER, ADDGROUP, ALTGROUP and LISTGRP commands.

VM retrieves a user's OpenExtensions information from RACF when that user logs on to the system. VM can also request OpenExtensions information from the

RACF database at any time. OpenExtensions VM services use this OpenExtensions information for the duration of the user's logon session.

RACF performs access checking for the byte file system (BFS) on VM. RACF and the BFS server perform the standard POSIX access checks, based on permission bits associated with a file in the BFS. RACF also audits file and directory access checks, as well as some other OpenExtensions events, based on LOGOPTIONS and AUDIT settings for several classes in the class descriptor table. These classes, which control the auditing characteristics in a POSIX environment, are: DIRSRCH, DIRACC, FSSEC, FSOBJ, and PROCESS.

Setting up Support for OpenExtensions

To set up RACF's support of OpenExtensions, perform these steps:

__ **1.** Define OpenExtensions information in your RACF USER profiles

See “Defining OpenExtensions Users” on page 241 for more information.

__ **2.** Define GIDs for your RACF GROUP profiles.

See “Defining OpenExtensions Groups” on page 245 for more information.

__ **3.** Identify your BFS service machine user IDs to RACF:

a. Create a profile called POSIXOPT.SETIDS in the VMPOSIX class.
b. Permit the BFS service machine user IDs.
c. If it is not active already, activate the VMPOSIX class.

Notes:

The BFS service machine must be logged off and then logged back on order for the change to take effect.

If you do *not* perform this step, the BFS service machine will incur an 0C6 abend when it invokes the RACF security exit. The following messages are part of the output that will be generated as a result of this abend:

```
DMS5SB2029I External security routine DMSSECIT called due to program check
RPICMS017I USER/RACF VM Racroute communication path has been terminated.
DMS5SB2029I External security routine DMSSECIT program check processing complete
```

See “OpenExtensions BFS File Security” on page 251 for more information about the BFS.

If you already have POSIXOPT SETIDS statements defined in the CP directory, you can run the RPIDIRCT EXEC to create the RACF commands that will define this profile and do the appropriate PERMITs. See “Using the RPIDIRCT EXEC” on page 248 for more information.

__ **4.** Decide if you want RACF to perform further authorization for executing set-UID and set-GID files in the BFS.

See “Protecting Set-UID and Set-GID Executable Files” on page 253 for details on protecting set-UID and set-GID files.

To use the RACF protection, perform these steps:

a. Create the appropriate profiles.
b. Permit users as appropriate.

c. If it is not active already, activate the VMPOSIX class.

By default, VM calls RACF for this authorization. RACF will fail the request if any of the following are true:

- The VMPOSIX class is not active
- The EXEC.U*uid* profile is not defined
- The EXEC.G*gid* profile is not defined.

Thus, by default, execution of set-id files will fail. If you do not want RACF to enforce the extra protection, turn off control for the DIAG280 event in your currently active VMXEVENT profile.

__ **5.** Determine which user IDs should be allowed to request OpenExtensions information about:

- Groups the user is not connected to
- Other users.

The user information which can be retrieved consists only of the OpenExtensions related data: the information from the user's OVM segment, and the user's supplementary group list. The group information which can be retrieved consists of a list of user IDs which are connected to the group. Field level access checking is not enforced by RACF when this information is requested.

On a POSIX system, this information is generally available to anyone. However, because users can't generally see this information using RACF commands, an optional mechanism is provided to restrict this ability.

By default, all OpenExtensions queries are allowed. To restrict this ability:

a. Create a profile called POSIXOPT.QUERYDB in the VMPOSIX class.
b. Permit only those users who need to make these queries.
c. If it is not active already, activate the VMPOSIX class.

It is not necessary to permit superusers to this profile.

In general, the OpenExtensions shell performs these queries on behalf of shell commands issued by users. For the most part, a simple mapping from a UID to a user ID, or a GID to a group name, is requested for display in the output of various commands such as ls. If the user is not authorized to retrieve the data, a UID or GID is displayed instead of the user ID or group name. As such, it is not considered a violation if the POSIXOPT.QUERYDB profile denies access to the requested information, and no violation message or audit record will be created.

If you already have POSIXOPT QUERYDB statements defined in the CP directory, you can run the RPIDIRCT EXEC to create the RACF commands that will define this profile and do the appropriate PERMITs.

Note: RPIDIRCT defines this profile with a universal access of READ and specifically denies access to users with a POSIXOPT QUERYDB DISALLOW statement in their directory entries.

See “Using the RPIDIRCT EXEC” on page 248 for more information.

__ **6.** Establish auditing options for OpenExtensions. See *RACF Auditor's Guide* for information on file-level and class-level auditing options.

__ **7.** Activate ESM support for BFS service machines.

To use RACF for BFS permission checking and auditing, you must do the following for each BFS server:

a. Specify ESECURITY as a start-up parameter in the DMSPARMS file. This file usually resides on the A-disk of each server and is named *server-id* **DMSPARMS**, where *server-id* is the user ID of the server. See *z/VM CMS File Pool Planning, Administration, and Operation* for more information.

b. Create or modify DMSESM PROFILE based on your installation needs.

   A file called DMSESM PROFILE tells the BFS file pool server which types of authorization checking will be routed to an external security manager (ESM). This step describes what this file should look like when RACF is the designated ESM for BFS.

   If your file pool server also contains SFS files and directories that are protected by RACF, see "Setting Up DMSESM PROFILE for RACF SFS Protection" on page 277.

   See *z/VM CMS File Pool Planning, Administration, and Operation* for details about each parameter specified in DMSESM PROFILE.

   Figure 19 shows the DMSESM PROFILE file shipped with z/VM.

```
P0 DMSSECIT
A0012 DMSUAUTH B0000 DMSAAUTH C00 DMSOAUTH D1 DMSSECIT
A002301 A002401
```

Figure 19. DMSESM PROFILE: Supplied with z/VM

   If you want RACF to protect BFS you must modify this file. To specify that RACF will be called for protection of BFS, add **E1 DMSPERM** to the end of line 2 in the DMSESM PROFILE supplied with z/VM. Figure 20 shows the updated file:

```
P0 DMSSECIT
A0012 DMSUAUTH B0000 DMSAAUTH C00 DMSOAUTH D1 DMSSECIT E1 DMSPERM
A002301 A002401
```

Figure 20. DMSESM PROFILE: Modified for RACF Protection of BFS

   Place the file on a disk that is accessible to each BFS server.

c. Replace the z/VM default BFS security exit with the exit provided by RACF.

   1) Copy ESMLIB CSLLIB from the RACF 305 disk to the BFS server's A-disk.

   2) Add the following command to the BFS server's PROFILE EXEC before the FILESERV START command:

      RTNLOAD DMSPERM (FROM ESMLIB

   See *CMS File Pool Planning, Administration, and Operation* for details on the external security manager exit.

d. Update the RACF SERVMACH file to identify the user ID of the RACF service machine that you want to be used for this BFS server.

All OpenExtensions audit records created by the BFS server are logged in the SMF DATA file of the specified RACF service machine. You can optionally dedicate a RACF service machine to a BFS server or group of BFS servers. See *RACF System Programmer's Guide* for information about dedicating a RACF service machine.

The RACF SERVMACH file is placed on the CMS Y-disk during RACF installation, and the default is to send OpenExtensions audit records to RACFVM. If you want OpenExtensions audit records logged on a RACF service machine other than RACFVM, copy the RACF SERVMACH file to another minidisk accessed by the BFS server and change the user ID to that of the RACF service machine desired.

__ **8.** Activate ESM support for the z/VM operating system.

To do this, specify your installation's value for NGROUPS_MAX in the ICHNGMAX macro in HCPRWA.

The POSIX constant NGROUPS_MAX defines how many GIDs will be associated with a process for the purpose of authority checking. The value you specify becomes the NGROUPS_MAX value for the duration of the IPL.

For instructions on coding the ICHNGMAX macro, see *RACF Macros and Interfaces*.

After you have updated ICHNGMAX and re-IPLed, VM will call RACF to:

- Retrieve OpenExtensions user and group information from RACF (rather than the CP directory):
  - When a user enters the system
  - In response to queries issued by CMS
- Authorize and audit changes to the identity of an OpenExtensions process
- Authorize and audit attempts to execute set-UID and set-GID files in the BFS.

Defining OpenExtensions Users

On an OpenExtensions system, users are defined to the system by a *user identifier* (UID). A *UID* corresponds to a VM user ID. POSIX allows for the definition of character representations of UIDs, called *user names,* to be associated with a UID. On z/VM, the user name associated with a UID is defined as its VM user ID in lower case. This is simply the convention VM uses in returning a user name to a program, or displaying the user name on the screen. There are no additional steps that need to be taken to define user names to either VM or RACF.

The same UID value can be assigned to multiple users (allowed by the POSIX standard), either explicitly or by default, but this is not recommended. If it is done, it should be done with care because the UID is used in OpenExtensions security checks and control at an individual user level would be lost. That is, users with the same UID value would be treated as a single user during POSIX security checks. In addition, when querying RACF for information associated with a particular UID, it is unpredictable which user name the returned information will be associated with.

The OpenExtensions information for a user is defined in the OVM segment of a USER profile. You can use the ADDUSER, ALTUSER, and LISTUSER commands to define and display OpenExtensions VM attributes for users in the OVM segment. Only users with SPECIAL authority or authority through field-level access checking are allowed to issue the commands specifying the OVM keyword.

For more information about field-level access checking, see "Setting Up Field-Level Access for the OVM Segment" on page 247.

The OpenExtensions attributes that can be defined in the OVM segment for a user are:

- UID: an integer value that identifies a user. If no UID value exists, RACF returns a value of 4 294 967 295 (X'FFFFFFFF') to VM. VM treats this value as the default UID. LISTUSER will return a value of **NONE** for this case.
- Home directory: the user's initial directory in the OpenExtensions file system. This becomes the current working directory for the user's process when he or she first uses an OpenExtensions function. If no home directory value exists, RACF defers to the value in the CP directory.
- Initial user program: the path name of the default program to be given control when a CMS user uses the OPENVM SHELL command to become an OpenExtensions user. It is usually a shell program. This program is not used when other non-OpenExtensions programs invoke an OpenExtensions service and become OpenExtensions processes. If no initial program value exists, RACF defers to the value in the CP directory.
- File system root: identifies the byte file system that contains the user's directory hierarchy. If no file system root value exists, RACF defers to the value in the CP directory.

RACF allows path name parameters to be up to 1023 mixed-case characters.

Assigning UIDs

The easiest way to assign UIDs to your RACF users is to run VM's DIRPOSIX utility against the CP directory, and then run RACF's RPIDIRCT EXEC. This will result in unique UIDs being assigned to every user ID.

For details on DIRPOSIX and RPIDIRCT, see "Using the RPIDIRCT EXEC" on page 248.

Using the ADDUSER Command

You can use the ADDUSER command to add a new OpenExtensions user. See *RACF Command Language Reference* for more information about the ADDUSER command.

ADDUSER Example

Suppose you want to add a new OpenExtensions user. The user profile will be owned by the system administrator's user ID, SYSADM, and will be a member of the existing group SYSOM which is associated with a GID. Enter:

```
ADDUSER PJWELLS DFLTGRP(SYSOM) OWNER(SYSADM) NAME('PATRICK J. WELLS')
 OVM(UID(27) HOME(/u/pjwells) PROGRAM(/bin/sh))
```

Using the ALTUSER Command

You can use the ALTUSER command to change information about an OpenExtensions user. See *RACF Command Language Reference* for more information about the ALTUSER command.

ALTUSER Example

To make existing OpenExtensions user CJWELLS a superuser and delete the PROGRAM from CJWELLS's profile so that the default shell program will be used when CJWELLS enters the OPENVM SHELL command, enter:

```
ALTUSER CJWELLS OVM(UID(0) NOPROGRAM)
```

Note: See *RACF Command Language Reference* for information you should consider before you change a UID.

Using the LISTUSER command

Use the LISTUSER command to list the details of the user profile. The details RACF lists from the OVM segment are the OpenExtensions user's:

- User identifier
- Initial directory path name
- Program path name
- File system root name.

LISTUSER Examples

To list OVM segment information for CJWELLS, enter:

```
LISTUSER CJWELLS OVM NORACF
```

The LISTUSER command displays:

```
USER=CJWELLS

OVM INFORMATION
----------------
 UID= 0000000024
 HOME= /u/cjwells
 PROGRAM= /u/cjwells/bin/myshell
 FSROOT= /../VMBFS:SERVER8:CJWELLS/
```

If the OVM segment does not exist, the LISTUSER command displays:

```
USER=CJWELLS

NO OVM INFORMATION
```

User Identity for an OpenExtensions Process

When a user ID logs on to VM, RACF performs user identification and authentication and, if successful, sends the OpenExtensions information contained within the user's USER profile back to CP, where the user's OpenExtensions security environment is maintained by z/VM. The UID from the OVM segment in the USER profile for the current user is used as the real, effective, and saved set-UID for the first OpenExtensions process created by the user.

OpenExtensions processes may have their UIDs changed during execution through the use of such system calls as **setuid()** and **seteuid()**. Any process subsequently created will inherit the UID values from its parent process. The effective UID of a process is used to make access check decisions for OpenExtensions resources. The VM user ID for the current user continues to be used to make access decisions for other VM resources.

Defining Superusers

POSIX 1003.1 defines a number of services that are allowed to pass security checks or perform special functions if the caller has appropriate privileges. In OpenExtensions, *appropriate privilege* is defined as superuser authority. A *superuser* is defined as a user with a UID equal to zero. A superuser is authorized to access all OpenExtensions protected objects and to issue virtually all security-relevant POSIX functions. The UID is not used when making existing (non-OpenExtensions) access decisions for VM resources (such as minidisks and SFS files) so the superuser privilege is confined to the OpenExtensions environment.

You should be very careful about who is given a UID of 0 as it allows almost unlimited access to OpenExtensions files and privileged functions.

Although assigning the same UID to multiple users is not generally recommended, it may be necessary in the case of superusers. You can quickly determine who your superusers are by looking at the access list of the mapping profile for UID 0:

```
RLIST VMPOSIX U0 ALL
```

For a detailed description of the mapping profiles and what they are used for, see “VMPOSIX Mapping Profiles for UIDs and GIDs” on page 254.

These mapping profiles should not be altered. Permitting a user to this profile will **not** result in a UID being assigned to the OVM segment of the USER profile.

The only OpenExtensions service that is not available to a superuser is the chaudit() (change audit) service when used to change the auditor audit options of a BFS file or directory. To change the auditor audit options, the user must have the RACF AUDITOR attribute. Superuser authority is not sufficient. A user can have both superuser and AUDITOR authority. Access to functions and resources by superusers are subject to auditing.

For details on auditing, see *RACF Auditor's Guide*.

Defining a Single Superuser

If your OpenExtensions environment is such that you never need more than one superuser logged on at a given time, consider defining a shared user ID for use as your superuser. For example:

```
ADDUSER SUPERUSR OVM(UID(0))
RDEFINE SURROGAT LOGONBY.SUPERUSR UACC(NONE)
PERMIT LOGONBY.SUPERUSR ID(userid) CLASS(SURROGAT) ACCESS(READ)
SETROPTS CLASSACT(SURROGAT)
```

where *userid* is a user who will LOGON to the superuser user ID using the BY option of the LOGON command.

See “Defining Shared User IDs” on page 69 for more information about shared user IDs.

Defining OpenExtensions Groups

On an OpenExtensions system, groups are defined to the system by a *group identifier* (GID). A *GID* corresponds to a RACF group. POSIX allows for the definition of character representations of GIDs, called *group names,* to be associated with a GID. In RACF, the group name is simply the name of the RACF GROUP profile which contains the GID. Although the CP directory in VM allows the definition of mixed-case group names, RACF GROUP profiles may only exist in upper case.

The same GID value can be assigned to multiple groups (allowed by the POSIX standard), either explicitly or by default, but this is not recommended. If it is done, it should be done with care because the GID is used in OpenExtensions security checks and control at an individual group level would be lost. That is, groups with the same GID value would be treated as a single group during OpenExtensions security checks. In addition, when querying RACF for information associated with a particular GID, it is unpredictable which group name the returned information will be associated with. The GID for a group is defined in the OVM segment of a GROUP profile. You can use the ADDGROUP, ALTGROUP, and LISTGRP commands to define and display the GID for a group. Only users with SPECIAL authority or authority through field-level access checking are allowed to issue the commands specifying the OVM keyword.

The OpenExtensions attributes that can be defined in the OVM segment for a group are:

- GID: an integer value that identifies a group. If no GID value exists, RACF returns a value of 4 294 967 295 (`X'FFFFFFFF'`) to VM. VM treats this value as the default GID.

Assigning GIDs

If you already have OpenExtensions groups defined in the CP directory, you can run the RPIDIRCT EXEC against it to create the equivalent RACF commands. You should consider this carefully before doing so. Unless the directory groups were defined with your existing RACF group structure in mind, then simply moving these group definitions into the RACF database may not have the results you desire. If it is not appropriate in your case to use RPIDIRCT, then you will need to manually assign GIDs to your RACF GROUPS, or write a program for this purpose. The output from the database unload utility may be helpful in identifying the RACF groups which are defined on the database.

For details on the database unload utility, see Chapter 9, “The RACF Database Unload Utility (IRRDBU00)” on page 133.

For details on RPIDIRCT, see “Using the RPIDIRCT EXEC” on page 248.

Using the ADDGROUP Command

You can use the ADDGROUP command to add a new OpenExtensions group. See *RACF Command Language Reference* for more information about the ADDGROUP command.

ADDGROUP Example

Suppose you want to add a new OpenExtensions group. The group profile will be owned by the system administrator's user ID, SYSADM. Enter:

```
ADDGROUP OVMIS4ME OWNER(SYSADM) OVM(GID(4))
```

Using the ALTGROUP Command

You can use the ALTGROUP command to change information about an OpenExtensions group. See *RACF Command Language Reference* for more information about the ALTGROUP command.

ALTGROUP Example

To give group VMIS4ME a new GID, enter:

```
ALTGROUP VMIS4ME OVM(GID(3243))
```

Using the LISTGRP Command

The LISTGRP command lists the details of the group profile. For the OVM segment, the LISTGRP command displays the OpenExtensions group's group identifier (GID).

LISTGRP Example

To request the listing of the OVM segment for group OVMIS4ME, enter:

```
LISTGRP OVMIS4ME OVM NORACF
```

The LISTGRP command displays:

```
INFORMATION FOR GROUP OVMIS4ME
  OVM INFORMATION
     GID= 0000003243
```

Group Identity for an OpenExtensions Process

When a user logs on to VM, RACF sends GID information contained within GROUP profiles back to the VM control program (CP). RACF returns the GID associated with the user's *primary group,* which is the user's default group as specified in the USER profile.

In addition to group names, OpenExtensions also allows for the definition of a *supplementary group* (SGID) list, which is a list of GIDs associated with the groups the user is connected to. The supplementary group list corresponds to list-of-groups processing in RACF. If SETROPTS GRPLIST is in effect, RACF returns the supplementary group list to CP, up to the value defined for NGROUPS_MAX on the ICHNGMAX macro in HCPRWA.

The GID from the OVM segment in the user's primary group is used as the real, effective, and saved set-GID for the first OpenExtensions process created by the user. OpenExtensions processes may have their GIDs changed during execution through the use of such system calls as **setgid()** and **setegid()**. Any process subsequently created will inherit the GID values from its *parent process*. The effective GID and supplementary group list are used to make access checking decisions for OpenExtensions resources.

Supplementary Group Lists

The supplementary group list is created only if SETROPTS GRPLIST is in effect, which means that list-of-groups processing is active in RACF. The supplementary group list cannot contain duplicate entries. If multiple groups share the same GID, which could be the default—4 294 967 295 (X'FFFFFFFF')—they are represented by a single entry. Thus, more groups than the value specified by NGROUPS_MAX can be represented in the supplementary group list.

These groups generally do not change until the user logs off and back on, though use of the **newgrp()** utility may result in a group being added to the SGID list.

Setting Up Field-Level Access for the OVM Segment

To let a user see or change OVM fields in a RACF user profile, you can set up field-level access. You can authorize a user to specified fields in the user's own profile. Generally speaking, you will want to prevent users from changing their UIDs, but will want to allow them to change their HOME, PROGRAM, and FSROOT fields. To authorize users to their OVM fields in this manner:

1. Define a profile for each of the OVM fields with a RACF RDEFINE command. Figure 21 shows commands for the four OpenExtensions fields.

```
RDEFINE FIELD USER.OVM.UID     UACC(NONE)
RDEFINE FIELD USER.OVM.HOME    UACC(NONE)
RDEFINE FIELD USER.OVM.PROGRAM UACC(NONE)
RDEFINE FIELD USER.OVM.FSROOT  UACC(NONE)
```

Figure 21. RDEFINE Commands for RACF OVM User Profile Fields

2. Permit users to access the fields with RACF PERMIT commands. Figure 22 shows commands for the four fields.
   - &RACUID lets all users look at their own fields.
   - READ access lets users read the UID field.
   - UPDATE access lets users change:
     - Their home directory in the HOME field
     - The program invoked for a VM OPENVM SHELL command in the PROGRAM field
     - Their file system root directory in the FSROOT field.

   Attention: Give only selected users UPDATE access to the UID field. A user with UPDATE access can become a superuser by changing the UID to 0.

```
PERMIT USER.OVM.UID     CLASS(FIELD) ID(&RACUID) ACCESS(READ)
PERMIT USER.OVM.HOME    CLASS(FIELD) ID(&RACUID) ACCESS(UPDATE)
PERMIT USER.OVM.PROGRAM CLASS(FIELD) ID(&RACUID) ACCESS(UPDATE)
PERMIT USER.OVM.FSROOT  CLASS(FIELD) ID(&RACUID) ACCESS(UPDATE)
```

Figure 22. PERMIT Commands for RACF OVM User Profile Fields

3. Activate the FIELD class with the RACF SETROPTS command; see Figure 23.

```
SETROPTS CLASSACT(FIELD)
```

Figure 23. SETROPTS Command

See *RACF Command Language Reference* for the other parameters on the RDEFINE, PERMIT, and SETROPTS commands.

Using the RPIDIRCT EXEC

The RPIDIRCT EXEC reads your CP directory and creates RACF commands that will populate the RACF database with the security-relevant information. See the *RACF Program Directory.* for more information about RPIDIRCT.

Note: It is recommended that you run the DIRPOSIX utility *before* you run the RPIDIRCT EXEC. DIRPOSIX, a sample utility provided by z/VM, assigns a unique UID to each user defined in the CP directory. See *z/VM Planning and Administration* for more information.

DIRPOSIX may also create new users and groups that must be changed. See “DIRPOSIX Considerations” on page 250 for more information.

z/VM includes a set of CP directory control statements that are used to define OpenExtensions information to VM. You can use the OVM option of the RPIDIRCT EXEC to process only the OpenExtensions information in the CP directory, as follows:

```
RPIDIRCT filename filetype filemode outmode (OVM
```

When you are satisfied with the output from RPIDIRCT, invoke RPIBLDDS as follows:

```
RPIBLDDS filename
```

where *filename* is the name of the SYSUT1 file created by RPIDIRCT.

If the OVM option is not specified, all of the RACF-related statements are processed, including the OpenExtensions statements.

Table 23 shows which RACF statements are generated for each of the OpenExtensions statements in the CP directory. See *z/VM: CP Planning and Administration* for descriptions of the CP directory control statements.

Table 23 (Page 1 of 2). RPIDIRCT Processing for OpenExtensions

For these CP directory statements:	**These RACF commands are generated:**
The first occurrence of: POSIXOPT QUERYDB in the CP directory	RDEFINE VMPOSIX POSIXOPT.QUERYDB UACC(READ)
POSIXOPT QUERYDB DISALLOW	PERMIT POSIXOPT.QUERYDB CLASS(VMPOSIX) ID(*userid*) ACCESS(NONE)
The first occurrence of: POSIXOPT SETIDs in the CP directory	RDEFINE VMPOSIX POSIXOPT.SETIDS UACC(NONE)
POSIXOPT SETIDs ALLOW	PERMIT POSIXOPT.SETIDS CLASS(VMPOSIX) ID(*userid*) ACCESS(READ)

Table 23 (Page 2 of 2). RPIDIRCT Processing for OpenExtensions

For these CP directory statements:	**These RACF commands are generated:**
POSIXINFO UID *uid*	ALTUSER user OVM(UID(*uid*))
POSIXINFO GNAME *gname*	CONNECT user GROUP(*gname*)
POSIXINFO IWDIR *string*	ALTUSER user OVM(HOME(*string*))
POSIXINFO IUPGM *string*	ALTUSER user OVM(PROGRAM(*string*))
POSIXINFO FSROOT *string*	ALTUSER *userid* OVM(FSROOT(*string*))
POSIXGROUP *gname gid*	ADDGROUP *gname* ALTGROUP *gname* OVM(GID(*gid*))
POSIXGLIST GNAMES *gname-1...gname-n*	CONNECT *userid* GROUP(*gname-1*) ⋮ CONNECT *userid* GROUP(*gname-n*)

Notes:

1. When the GIDS parameter is used on the POSIXGLIST statement, or the GID parameter is used on the POSIXINFO statement, the GID is changed to its corresponding group name (*gname*) for use in the RACF CONNECT command.

2. Although VM supports mixed-case group names, RACF does not. Thus, if your CP directory contains the following two statements:

   ```
   POSIXGROUP Payroll 10
   POSIXGROUP payroll 20
   ```

   RACF will only create the PAYROLL group profile, assigning it a GID value of 10. A warning message will be issued when RPIDIRCT encounters the second POSIXGROUP statement.

3. RACF does not allow a profile in the GROUP class to have the same name as a profile in the USER class. Thus, you cannot have a group name on a POSIXGROUP statement that is the same as any user ID on a USER statement. (RACF first converts the group name to uppercase.) RPIDIRCT does not issue any error messages in this situation, but, when the RPIBLDDS EXEC executes the commands generated by RPIDIRCT, either the ADDUSER or the ADDGROUP will fail, depending on the order of execution.

4. For each *gname* on a POSIXGROUP statement, RPIDIRCT first generates an ADDGROUP statement to define the group, and then an ALTGROUP statement to add the GID to the group. It is done this way in case you intentionally used existing RACF group names on POSIXGROUP statements. If a single ADDGROUP command had been generated, it would fail, and a GID would not be assigned to the group. If you did use existing RACF group names, you can choose to remove the ADDGROUP commands from the output file, or simply let them run and they will fail without any harm.

5. POSIXOPT EXEC_SETIDS statements will not result in the creation of any RACF commands to create or modify EXEC.U*uid* or EXEC.G*gid.* profiles in the VMPOSIX class. There is no one-to-one correspondence between the CP directory and RACF support of executing set-UID and set-GID files in the byte file system, and careful consideration should be given in implementing these controls. See “Protecting Set-UID and Set-GID Executable Files” on page 253 for details on protecting set-UID and set-GID files.

6. A user's default RACF group is not changed as a result of any POSIXINFO or POSIXGLIST statements.
7. OpenExtensions defaults established in the system configuration file are not processed by RPIDIRCT.
8. If you used DIRPOSIX with the SYSENTRIES option, your CP directory will contain a reference to the implicitly-defined group called **DEFAULT**. RPIDIRCT will issue a message referring to this undefined group. This is expected and can be ignored.

DIRPOSIX Considerations

In addition to adding UID values for all defined users in your CP directory, DIRPOSIX also defines users and groups that are common on many UNIX systems. However, RACF cannot support all the entries as created by the SYSENTRIES option for the following reasons:

- Certain UID and GID values will not be valid on RACF ADDUSER and ALTUSER commands (for example: 4294967294 and 4294967295).
- Some group names are identical to user names (**bin** and **sys**, for example). RACF does not allow both a user profile and a group profile with the same name.

RPIDIRCT allows for these differences by providing a control file, RPIDIRCT CNTRL, which contains a set of UID, GID, and group name values that are acceptable to RACF. RPIDIRCT substitutes these values for the SYSENTRIES default values in the output file created by RPIDIRCT. Some of the names are referenced in other OpenExtensions configuration files.

The BFS LOADBFS configuration file is used to install the byte file system. The file is used once during installation and is not referenced again. See *z/VM CMS File Pool Planning, Administration, and Operation* for more information.

The SHELL LOADBFS configuration file is used to install the optional Shell and Utilities feature, and is also used every time the Shell and Utilities are serviced. This file is owned by the VMSES/E user ID that is used to install the Shell and Utilities feature. Both of these files refer to users **bin** and **root** and groups **bin** and **system**. If you implement RACF protection of OpenExtensions *after* OpenExtensions for z/VM has been installed, file ownership is already established in the byte file system. It is best not to change this ownership. Within the file system, ownership is based on the UID and GID values, not the user name and group name values.

When you implement RACF protection, it is recommended that you do the following:

1. Run the DIRPOSIX utility with the SYSENTRIES option.
2. Run the RPIDIRCT utility to automatically change the UID and GID values according to the options set in the RPIDIRCT CNTRL file.

   Any conflicting group names are changed to the group names shown in Table 24 on page 251.

   RPIDIRCT leaves any GID or UID of 4294967295 as undefined. By convention, if a group or user is defined but does not have an OVM segment, RACF assumes a GID or UID of 4294967295.

See Table 24 on page 251 for the changes made by RPIDIRCT. You can modify RPIDIRCT CNTRL if the names are not acceptable at your installation. The syntax of the control statements is documented in the file.

3. Replace all occurrences of the group **bin** in BFS LOADBFS (and in SHELL LOADBFS if you have installed the optional OpenExtensions Shell & Utilities Feature) with **GBIN**.

   If you have modified RPIDIRCT CNTRL to use a value other than **GBIN**, you must make the equivalent change in BFS LOADBFS (and SHELL LOADBFS).

Table 24 maps the values added to the CP directory by the DIRPOSIX SYSENTRIES option to values that are compatible with RACF.

Table 24. RPIDIRCT CNTRL Changes to DIRPOSIX SYSENTRIES

DIRPOSIX Default Value	RACF-Compatible Value
system	SYSTEM
staff	STAFF
bin	GBIN
sys	GSYS
adm	GADM
mail	MAIL
security	SECURITY
nobody	GNOBODY
4294967294	2147483647

OpenExtensions BFS File Security

In z/VM, the POSIX file system is called the *byte file system (BFS).* The BFS is implemented in OpenExtensions VM as an extension of the *shared file system (SFS).*

The files within the file systems are managed by the BFS server. Security information is carried along with the file rather than in a RACF profile. Security information consists of such items as the file owner UID and GID, the file's permission bits, and audit settings. BFS calls an ESM exit to perform access decisions. The access rules RACF that enforces are POSIX 1003.1 access controls, which use permission bits.

There are three different sets of users who might want to access a file:

- The owner of the file
- The members of the owning group
- All other users on the system

These users might want four different kinds of access permission (of which only three apply to any given file):

- Read
- Write
- Search (directories only)
- Execute (files other than directories)

OpenEdition VM protects its data by means of file *permission bits.* Every file has a three-part string of file permission bits associated with it. The three parts represent the owner, group, and others, and one bit within each part represents each type of access. The owner of the file sets the bits on or off to grant or deny permission to access the file. RACF recognizes these bits and grants or denies access accordingly.

The ESM exit is also called to authorize various functions that alter security information for a file. For example, if a user wishes to change the owner of a file, or the permission bits associated with a file, the ESM exit running in BFS will be called to authorize the function (and perform auditing if necessary).

The permission bits can be changed using the **chmod** command. The file owner UID can be changed using the **chown()** command and the file owner GID can be changed using the **chgrp()** command. For a general discussion on file security, see *OpenExtensions for z/VM User's Guide*. For information about OpenExtensions commands, see *OpenExtensions for z/VM Command Reference*.

The authorization checks that the BFS server makes in the absence of an external security manager (ESM) are documented in *z/VM CMS File Pool Planning, Administration, and Operation*. For the most part, RACF enforces the same rules, because the POSIX standard is being enforced. There are some exceptions.

- BFS allows the owner and auditor audit options contained within a BFS file or directory to be modified, but does not produce any audit records as a result of the audit options. RACF will produce audit records based on the file level audit options, depending on the class wide settings for the RACF OpenExtensions auditing classes (see *RACF Auditor's Guide*).
- RACF allows a user with the RACF AUDITOR attribute to enforce the installation's auditing policy in the OpenExtensions environment. A RACF auditor can modify the auditor audit options in any BFS file or directory. Further, a RACF auditor has automatic READ and SEARCH access to any BFS directory; authority through the permission bits associated with the directory is not necessary.
- RACF allows a further level of protection for set-UID and set-GID files beyond requiring execute access to the file through the file permission bits. See "Protecting Set-UID and Set-GID Executable Files" on page 253 for details.
- BFS considers a user with SFS administrator authority to have appropriate privileges. That is, an SFS administrator is considered to be a superuser. RACF does not consider an SFS administrator to be a superuser.

Changing the Identity of an OpenExtensions Process

In POSIX, there are defined methods of changing the UIDs and GIDs (real, effective, and saved set) of a running process, which means it will run with different access authorities. The functions setuid() and seteuid() can change UIDs associated with a process. Similarly, setgid() and setegid() can change the GIDs associated with a process, as can the newgrp() utility. This could be useful for example to make it possible for an authorized ***daemon*** process to create processes which are associated with other login UIDs.

The POSIX standard defines what authority is required to perform these privileged functions, and the results the function should have upon successful completion.

For example, setuid() changes the effective UID of a process to the one specified only if the requesting UID is a superuser, or if the specified UID is the same as the real or saved set-UID of the process. Further, if the requester is a superuser, then all three of the process' UIDs (real, effective and saved set) are changed to the value specified. This would affect the way subsequent setuid() and seteuid() calls work.

RACF enforces the rules of the POSIX standard for these functions, just as VM would in the absence of an ESM. For newgrp, which is an extension of the POSIX standard, RACF will allow a user to switch to the new group if the user is connected to the GROUP profile in the RACF database. RACF will also audit these functions depending on the SETROPTS LOGOPTIONS setting for the PROCESS class. See *RACF Auditor's Guide* for details.

z/VM will not call RACF to authorize or audit changes to the identity of an OpenExtensions process unless the ICHNGMAX macro in HCPRWA has been coded with a valid NGROUPS_MAX value. See “Setting up Support for OpenExtensions” on page 238 for details.

Protecting Set-UID and Set-GID Executable Files

A *set-UID* file is a BFS executable file that changes the effective UID of the process running the program to the owning UID of the file. A *set-GID* file is a BFS executable file that changes the effective GID of the process running the program to the owning GID of the file.

A superuser or the file owner can use a **chmod** shell command or **chmod()** function to change an executable file into a set-UID and/or set-GID file by turning on bits in the file mode of the file.

If one or both of these bits are *on*, the effective UID, effective GID, or both, plus the saved UID, saved GID, or both, for the process running the program are changed to the owning UID, GID, or both, for the file. This change temporarily gives the process running the program access to data the file owner or group can access.

In a new file, both bits are set *off*. Also, if the owning UID or GID of a file is changed or if the file is written in, the bits are turned *off*.

As with any other executable file within BFS, a set-UID or set-GID program may not be run for a user unless that user has execute access to the file containing the program. For details on file permissions, see “OpenExtensions BFS File Security” on page 251.

When a user executes a set-UID or set-GID file, the file is copied to the user's address space for execution. As such, the user has the ability to trace and modify the program as it is executing. The user would also then be able to pass control off to another program which would then continue to execute under the authority of the original file owner. Such a program would be able to access BFS files based on the file owner's access privileges. In this regard, set-UID and set-GID files create security-sensitive processes on z/VM.

To further restrict the ability to execute set-UID and set-GID files beyond requiring execute access to the file, z/VM provides the POSIXOPT EXEC_SETIDS ALLOW directory control statement to specify which users are allowed to have their effective

UID/GID changed during exec(). This statement would be placed into the directory entry of any user you want to be able to execute **all** set-UID and set-GID files for which they have execute permission (note that the system configuration file also contains a statement which indicates the system-wide default value to be used if a user's directory entry does not contain a POSIXOPT EXEC_SETIDS statement).

RACF will allow a further level of granularity to protect exec() processing beyond the all-or-none ability of POSIXOPT EXEC_SETIDS ALLOW. Define profiles in the VMPOSIX class of the format EXEC.U*uid* and EXEC.G*gid*, where *uid* is the file owner's UID and *gid* is the file owner's GID. A user with READ access to this profile (or these profiles) is authorized to execute a program running under the UID and/or GID of the owner of the program (depending on whether the set-UID bit, set-GID bit, or both, is set in the file mode of the file being executed). For example, to allow user ID SUSAN to execute a file which will run with a UID of 30:

```
  RDEFINE VMPOSIX EXEC.U30 UACC(NONE)
  PERMIT EXEC.U30 CLASS(VMPOSIX) ACCESS(READ) ID(SUSAN)
```

The VMPOSIX profiles are checked **in addition** to the check for execute access to the file. It is not necessary to permit superusers to these profiles. If the profile does not exist, or the VMPOSIX class is not active, RACF will fail the exec().

In order for VM to give control to RACF to authorize the exec of a set-UID or set-GID file in BFS, control must be on in your VMXEVENT profile for the DIAG280 event, which is the mechanism by which exec is implemented on VM. By default, DIAG280 will be controlled when you install RACF on z/VM. If you do not wish to enforce exec protection in the VMPOSIX class, then turn off control for DIAG280 in your VMXEVENT profile.

```
RALTER VMXEVENT profile-name ADDMEM(DIAG280/NOCTL)
SETEVENT REFRESH profile-name
```

This will avoid the performance overhead associated with the call to RACF. If control is off for DIAG280, VM will check the POSIXOPT settings in the CP directory to determine if the user is authorized to execute the set-UID or set-GID file. For details on how to control events in your VMXEVENT profile, see “Creating VMXEVENT Profiles” on page 153..

These VMPOSIX profiles provide the ability to restrict execute access to set-UID and set-GID files in BFS at an individual user or group level. In contrast, using only the ***other*** execute access permission bits to restrict usage of the set-UID or set-GID file corresponds to using UACC(READ) in a RACF profile. Also, the ability to execute set-UID files may be separated from the ability to execute set-GID files. For details on file permissions, see “OpenExtensions BFS File Security” on page 251.

VMPOSIX Mapping Profiles for UIDs and GIDs

For each UID that has been defined in the OVM segment of a USER profile, a VMPOSIX profile called U*uid* exists. The access list of a U*uid* profile contains all user IDs that have been assigned this UID.

For each GID that has been defined in the OVM segment of a GROUP profile, a VMPOSIX profile called G*gid* exists. The access list of a G*gid* profile contains all groups that have been assigned this GID.

These mapping profiles are used to provide a cross-reference to USER and GROUP profiles. They provide RACF with a performance sensitive method of returning information for a given UID or GID when requested by z/VM or application programs.

RACF maintains these mapping profiles automatically when UIDs and GIDs are added, changed, or deleted.

For example, if the following command is issued

```
  ADDUSER BRUCE OVM(UID(13))
```

RACF creates a VMPOSIX profile named U13 with BRUCE contained on the access list. If the following command is subsequently issued

```
  ALTUSER BRUCE OVM(UID(55))
```

RACF deletes the U13 profile and creates a U55 profile with BRUCE contained on the access list.

In general, *you should not alter these profiles.* However, it is possible they might get inadvertently deleted, or damaged by database corruption. If a profile is deleted, or if the user is not contained in its access list, VM will not be able to retrieve information for the UID or GID that the profile represented. RACF will be unable to locate the mapping profile and will send VM a return code indicating that the UID or GID is invalid.

If this happens, an authorized user needs to repair the damage. First, see if the user name associated with the UID or the group name associated with the GID can be determined from a message displayed by OpenExtensions VM. For example, if the user name is BRUCE, enter:

```
LISTUSER BRUCE OVM NORACF
```

to display the UID associated with BRUCE. If, for example, LISTUSER displays a UID of 13, you would then enter:

```
  RDEFINE VMPOSIX U13 UACC(NONE)
  PERMIT U13 CLASS(VMPOSIX) ACCESS(NONE) ID(BRUCE)
  PERMIT U13 CLASS(VMPOSIX) ID(your-userid) DELETE
```

The second PERMIT command is necessary because RDEFINE puts the profile creator on the access list.

If you are unable to determine the user name or group name from VM, look at the output from the database unload utility to find the user ID or group associated with a given UID or GID. The mapping profiles should then be added, changed or deleted as appropriate.

Chapter 15. Protecting the VM Shared File System (SFS)

RACF provides security for the VM shared file system (SFS) by acting as the external security manager (ESM) for SFS. As documented in *z/VM: CMS File Pool Planning, Administration, and Operation,* an ESM can augment or replace the security provided with SFS on z/VM.

RACF provides the ability to protect the following items in SFS:

- SFS directories
- SFS files
- SFS external objects
- SFS administrator commands
- SFS operator commands

This chapter includes the following topics:

- Controlling access to SFS files and directories:

  General information you should know about protecting SFS files and directories.

- Implementing RACF protection for SFS files and directories:

  Steps for implementing RACF protection for SFS files and directories

- Controlling the use of SFS administrator and operator commands:

  General information and implementation steps for protecting SFS administrator and operator commands

- Setting up DMSESM PROFILE for RACF SFS protection:

  Plan which settings you need in the DMSESM PROFILE to specify how SFS will call RACF.

- Activating RACF as the SFS external security manager:

  Steps for activating the RACF protection of SFS.

- SFS administration with RACF:

  RACF administration topics associated with SFS.

- End user interaction with SFS and RACF:

  RACF SFS file and directory commands that replace SFS functions.

Controlling Access to SFS Files and Directories

You can activate the FILE and DIRECTRY classes to protect files and directories in the shared file system (SFS). Using the FILE and DIRECTRY profiles, users can provide specific protection for specific files or directories, and, through the use of generic profiles, for sets of files and directories.

Some reasons for using the FILE and DIRECTRY classes provided by RACF:

- The same RACF auditing and reporting is available for FILE and DIRECTRY profiles, as for other RACF resource profiles. For example, you can audit accesses to SFS files and directories and changes to the profiles protecting them.
- You can specify a NOTIFY user ID to be notified of failed access attempts.
- You can use generic RACF profiles to protect more than one file or directory the same way. For example, you can protect:
  - All files in a specific directory, and any of its subdirectories. This includes all existing files and any that are added in the future.
  - All files of a specific file type (SCRIPT, for example) in a specific directory. This includes all existing files of the specified type and any of that type that are added in the future.
  - All subdirectories in a specific directory, and any of their subdirectories.
- If the SETROPTS GENERICOWNER option is in effect on your system, you can control who can create profiles by specifying the user ID that appears in the profile name on the OWNER operand.
- If security classifications (such as security labels) are used on your system, they can be specified in FILE and DIRECTRY profiles.

Working With FILE and DIRECTRY Profiles

To work with SFS files and directories, you should use the RACF commands provided specifically for them:

Table 25. RACF Commands Used to Work with FILE and DIRECTRY Profiles

Activity	**FILE Profiles**	**DIRECTRY Profiles**
Defining	ADDFILE	ADDDIR
Changing	ALTFILE	ALTDIR
Listing	LFILE	LDIRECT
Granting or denying access	PERMFILE	PERMDIR
Searching	SRFILE	SRDIR
Deleting	DELFILE	DELDIR

For the authority needed to issue any of these commands, see *RACF Command Language Reference*.

RACF SFS Command Examples

- Add a FILE profile

  ```
  RAC ADDFILE CHECK SCRIPT POOL2:ANDREW.PAYROLL UACC(NONE) NOTIFY(ANDREW)
  ```

- Alter a DIRECTRY profile

  ```
  RAC ALTDIR POOL2:ANDREW.PAYROLL SECLABEL(SECRET)
  ```

- Delete a FILE profile

  ```
  RAC DELFILE PAY1994 LIST3820 SERVER2:ANDREW.
  ```

- List a DIRECTRY profile

  ```
  RAC LDIRECT POOL2:ANDREW.PAY1994.TAX.DEDUCT AUTHUSER HISTORY
  ```

- Authorize another user to read your SFS files

  ```
  RAC PERMFILE * * SERVER2:ANDREW.** ID(LAURIE) ACCESS(READ)
  ```

- Copy an access list from a FILE profile to a DIRECTRY profile

  ```
  RAC PERMDIR SERVER2:ANDREW.** FCLASS(FILE) FROM(* * SERVER:LAURIE.**) RESET
  ```

- Search for DIRECTRY profiles

  ```
  RAC SRDIR LEVEL(10) FILTER(POOL2:ANDREW.**.DEDUCT)
  ```

- Search for FILE profiles

  ```
  RAC SRFILE SECLABEL(SECRET) FILTER(* * SERVER2:ANDREW.PAY*.**)
  ```

RACF Format for SFS Directory and File Names

Profile names for SFS objects (FILE and DIRECTRY classes) are stored in the RACF database in a different format than when they are entered using the RACF SFS commands. When using these commands, RACF follows the SFS naming conventions.

The **SFS format** of an SFS directory name is:

file-pool-id:*userid.dir1.dir2...*

The **SFS format** of an SFS file name is:

filename filetype directory-id

or:

filename filetype file-pool-id:*userid.dir1.dir2...*

To make authority checking more efficient, RACF converts the SFS file or directory name to a RACF format.

The **RACF format** of SFS directory names is:

file-pool-id.userid.dir1.dir2

The **RACF format** of SFS file names is:

file-pool-id.userid.dir1.dir2.filename.filetype

Table 26 on page 261 shows some profile name examples.

Table 26. RACF and SFS Profile Name Formats

Type	SFS Format	RACF Format
Directory	FP1:OPERATOR.DIR1.DIR2.DIR3	FP1.OPERATOR.DIR1.DIR2.DIR3
Directory	FP1:OPERATOR.	FP1.OPERATOR
File	ONE SCRIPT FP2:OPER.DIR1.DIR2	FP2.OPER.DIR1.DIR2.ONE.SCRIPT
File	MY SCRIPT FP2:OPER.	FP2.OPER.MY.SCRIPT

The SFS format of FILE and DIRECTRY profile names is used in the:

- RACF SFS file commands (ADDFILE, ALTFILE, DELFILE, LFILE, PERMFILE, and SRFILE)
- RACF SFS directory commands (ADDDIR, ALTDIR, DELDIR, LDIRECT, PERMDIR, and SRDIR)
- Output of the LDIRECT, LFILE, SRDIR, and SRFILE commands.

The RACF format of FILE and DIRECTRY profile names is:

- Used to store the profile name in the RACF database
- Used to define an entry for the FILE or DIRECTRY class in the global access checking table
- Used to process FILE and DIRECTRY profiles in RACF commands other than the RACF SFS commands (RALTER, RDEFINE, and RLIST, for example)
- Used on the RACROUTE macro when specifying a resource name (ENTITY and ENTITYX keywords, for example)
- Used in audit records
- Displayed in reports produced by such RACF utilities as DSMON, RACUT200, and the RACF report writer
- Displayed in the output of the RACF database unload utility and the SMF data unload utility.

For more information about specifying FILE and DIRECTRY profile names, see *RACF Command Language Reference*.

Protecting SFS External Objects

RACF protects SFS external objects as resources in the FILE class. An *SFS external object* is an SFS directory entry. This entry contains a *remote name*, which is the name of an object that is not managed by the local SFS server. An SFS external object might not be an “object” at all; it is a character string stored in SFS catalogs and its use is defined by the application.

For example, if you have an SFS external object named EXTOBJ1 INFO in your FPOOL1:ANDREW.DEPT22 directory, the RACF resource name in the FILE class is:

```
EXTOBJ1 INFO FPOOL1:ANDREW.DEPT22
```

Use the RACF SFS file commands (ADDFILE, ALTFILE, DELFILE, LFILE, PERMFILE, and SRFILE) to create, update, and delete profiles for SFS external objects.

If there is no RACF profile protecting an SFS external object, access to the external object will be denied.

Access Authority for SFS Files and Directories

SFS files and directories on VM can have one of these access authorities:

NONE The user or group is denied access to the SFS file or directory.

> **Attention**
>
> Anyone who has READ, UPDATE, CONTROL, or ALTER authority to a protected SFS file or directory can create copies of the data in them. If a user copies the data files to an SFS file or directory for which he or she can control the security characteristics, the user can downgrade the security characteristics of the copied files. For this reason, you will probably want to assign a UACC of NONE, and then selectively permit a small number of users to access your SFS file or directory, as their needs become known. (See *RACF General User's Guide* for information on how to permit selected users or groups to access an SFS file or directory.)

READ The user or group is authorized to access the SFS file or directory for reading only.

UPDATE The user or group is authorized to access the SFS file or directory for reading or writing only.

CONTROL Equivalent to UPDATE.

ALTER Lets users read, update, erase, discard, rename, or relocate the SFS file or directory.

When ALTER is specified in a:

- Discrete profile, users can read, alter, and delete the profile itself, *including the access list*. However, ALTER does not allow users to change the owner of the profile.
- Generic profile, users have *no* authority over the profile itself.
- Generic DIRECTRY profile, users can create SFS directories protected by the profile.
- Generic FILE profile, users can create SFS files protected by the profile.

Note: The actual access authorities required for specific SFS operations depends on the operation itself. Multiple authorities might be required.

How SFS Interacts with RACF Profiles and Authorizations

When RACF provides security for SFS, many authorities that are required for SFS functions without RACF are also required with RACF. For example, file or directory ownership is required for some commands.

If SFS requires read authority to a file or directory for a specific function, RACF READ authority is required. If SFS requires read/write authority for a function, RACF UPDATE authority is required. For example, to change or edit an existing SFS file in a FILECONTROL or DIRCONTROL directory, you must have UPDATE access to the FILE profile that protects the SFS file.

RACF profiles are not created automatically as a result of the RACROUTE interface between SFS and RACF, but profiles may be deleted or renamed. These cases are noted in the following list.

Attention: If there is no RACF profile protecting an SFS file, directory, or external object, access to the resource will be denied.

Here is a list of common SFS functions and their corresponding required RACF authorities.

1. Read a file in a FILECONTROL or DIRCONTROL directory

   RACF authorization required:

   - READ access to the file

2. Write to an existing file in a FILECONTROL or DIRCONTROL directory

   RACF authorization required:

   - UPDATE access to the file

3. Read a directory (using LISTDIR, for example)

   RACF authorization required:

   - READ access to the directory

4. Create a directory

   RACF authorization required:

   - UPDATE access to the parent directory
   - ALTER access to the new directory name

     Note: This does not create a discrete DIRECTRY profile, but rather it checks your access to an existing DIRECTRY profile. If no profile exists to protect the new directory, the request will be denied. You must then create a DIRECTRY profile to protect the directory, and reissue the command to create the directory.

5. Create a file or external object in a FILECONTROL or DIRCONTROL directory

   RACF authorization required:

   - UPDATE access to the parent directory
   - ALTER access to the new file or external object name

     Note: This does not create a discrete FILE profile, but rather it checks your access to an existing FILE profile. If no profile exists to protect the new file or external object, the request will be denied. You must then create a FILE profile to protect the file or external object, and reissue the command to create the file or external object.

6. Create an alias in a FILECONTROL directory

   RACF authorization required:

   - READ access to the base file.
   - UPDATE access to the target directory where the alias will reside.
   - The target directory's owner (if different from the command issuer) must have READ access to the base file.

Note: No RACF checking is performed on the name of the alias itself. An alias is protected by the profile which protects the base file.

7. Erase or discard a directory

   RACF authorization required:

   - UPDATE access to the parent directory
   - ALTER access to the directory being deleted

   After SFS has deleted the directory, RACF is called to also delete RACF protection from the directory. If a discrete profile is protecting the directory, the DIRECTRY profile will be deleted. If a generic profile is protecting the directory, the profile is not deleted, but an audit record may be written to record the directory deletion.

   Note: If the FILES option is used to erase or discard a directory that contains files, authorization to erase or discard each file is also required. (See the next list item.)

8. Erase or discard a file or external object in a FILECONTROL or DIRCONTROL directory

   RACF authorization required:

   - UPDATE access to the parent directory
   - ALTER access to the file or external object being deleted

   After SFS has deleted the file or external object, RACF is called to also delete RACF protection from the file or external object. If a discrete profile is protecting the file or external object, the FILE profile will be deleted. If a generic profile is protecting the file or external object, the profile is not deleted, but an audit record may be written to record the file or external object deletion.

9. Erase or discard an alias in a FILECONTROL directory

   RACF authorization required:

   - UPDATE access to the parent directory
   - READ access to the base file

10. Relocate a file or external object in a FILECONTROL directory

    RACF authorization required:

    - ALTER access to the file or external object being relocated

    - UPDATE access to the parent directory

    - UPDATE access to the target parent directory

    - Authority to create the new FILE profile (same as requirements for issuing ADDFILE) if a discrete profile exists for the old file, *or,* ALTER access to the generic profile that will protect the new file if a generic profile was protecting the old file

    After SFS has relocated the file, RACF is called to also *relocate* RACF protection for the file. If a discrete profile is protecting the file, the FILE profile will be renamed. If a generic profile is protecting the file, the profile is not deleted, but an audit record may be written to record the file being renamed.

11. Relocate an alias in a FILECONTROL directory

    RACF authorization required:

    - UPDATE access to the parent directory

- UPDATE access to the target parent directory
- READ access to the base file

12. Rename a file or external object in a DIRCONTROL or FILECONTROL directory

    RACF authorization required:

    - ALTER access to the file or external object being renamed
    - UPDATE access to the parent directory
    - Authority to create the new FILE profile (same as requirements for issuing ADDFILE) if a discrete profile exists for the old file, *or,* ALTER access to the generic profile that will protect the new file if a generic profile was protecting the old file

    After SFS has renamed the file, RACF is called to also *rename* RACF protection for the file. If a discrete profile is protecting the file, the FILE profile will be renamed. If a generic profile is protecting the file, the profile is not deleted, but an audit record may be written to record the file being renamed.

13. Rename an alias in a FILECONTROL directory

    RACF authorization required:

    - READ access to the base file
    - UPDATE access to the parent directory

14. Protecting a DIRCONTROL directory and its files

    RACF has no knowledge of which directories are DIRCONTROL directories; therefore, RACF treats them the same. RACF profiles in the DIRECTRY and FILE classes protect objects in a DIRCONTROL directory just as they do in a FILECONTROL directory. Common SFS functions related to DIRCONTROL and FILECONTROL directories and their RACF authorization requirements are shown above.

    RACF does not support the SFS authorities DIRREAD and DIRWRITE, but generic profiles may be used to provide the same support.

    - DIRREAD authority

      DIRREAD authority allows a user to read the directory, all files in the directory, and all files added in the future. To protect a directory in this way with RACF, you need two profiles:

      a. A DIRECTRY profile which protects the directory. For example:

         ```
         RESEARCH:USER1.DIRC
         ```

      b. A FILE profile which protects all of the files in the directory. For example:

         ```
         * * RESEARCH:USER1.DIRC
         ```

      If you gave USER2 READ access to both of these profiles, then USER2 would have the equivalent of DIRREAD access. This is true only if you do not create more specific profiles for individual files within the directory, such as:

      ```
      *       SCRIPT RESEARCH:USER1.DIRC
      PRIVATE EXEC   RESEARCH:USER1.DIRC
      ```

Once you create these more specific FILE profiles, USER2 may no longer have the equivalent of DIRREAD authority.

- DIRWRITE authority

  DIRWRITE authority allows a user to read from and write to the directory, all files in the directory, and all files added in the future. To protect a directory in this way with RACF, you need two profiles as listed in the examples above.

  If you gave USER2 UPDATE access to both of these profiles, then USER2 would have the equivalent of DIRWRITE access. This is true only if you do not create more specific profiles for individual files within the directory, as described for DIRREAD.

Restrictions for SFS Directory RENAME and RELOCATE

The SFS functions for renaming a directory and relocating a directory involve renaming many subdirectories and underlying files in one command. The RACROUTE interface between SFS and RACF does not support these multiple renames/relocates. When a directory is relocated or renamed, a RACROUTE call is generated to rename the directory itself, but no calls are generated to rename underlying subdirectories and files. So if you have discrete profiles protecting files or directories which are in the structure beneath the directory being renamed or relocated, these profiles will not be renamed and will no longer protect their corresponding objects. Also, if you have generic profiles protecting only files or directories which are in the structure beneath the directory being renamed or relocated (and they protect no files and directories in the structure above it), these profiles will not be renamed and will no longer protect any objects.

To prevent this situation from occurring, the RENAME DIRECTORY and RELOCATE DIRECTORY commands should not be used when RACF is being used to protect SFS files and directories. The use of these commands can be controlled using SFSCMD profiles beginning with RENAME and RELOCATE. The RACF administrator should create profiles as follows:

```
RAC SETROPTS GENERIC(SFSCMD) CLASSACT(SFSCMD)

RAC RDEFINE SFSCMD RENAME.** UACC(NONE)
RAC PERMIT RENAME.** CLASS(SFSCMD) RESET

RAC RDEFINE SFSCMD RELOCATE.** UACC(NONE)
RAC PERMIT RELOCATE.** CLASS(SFSCMD) RESET
```

Instead of using the RENAME DIRECTORY and RELOCATE DIRECTORY commands, the ICHDIRMV EXEC may be used. See “Using the ICHDIRMV EXEC” on page 267 for more information.

If the RENAME DIRECTORY and RELOCATE DIRECTORY commands are not protected or allowed only for certain users, the following list describes the RACF authorizations required.

1. Relocate a directory

   RACF authorization required:

   - UPDATE access to the SFSCMD profile that protects the RELOCATE command
   - UPDATE access to the parent directory

- ALTER access to the directory being relocated
- UPDATE access to the target parent directory
- Authority to create the new DIRECTRY profile (same as requirements for issuing ADDDIR)

After SFS has relocated the directory, RACF is called to also *relocate* the directory. If a discrete profile is protecting the directory, the DIRECTRY profile will be renamed. If a generic profile is protecting the directory, the profile is not deleted, but an audit record may be written to record the directory being renamed.

2. Rename a directory

   RACF authorization required:

   - UPDATE access to the SFSCMD profile which protects the RENAME command
   - UPDATE access to the parent directory
   - ALTER access to the directory being renamed
   - Authority to create the new DIRECTRY profile (same as requirements for issuing ADDDIR)

   After SFS has renamed the directory, RACF is called to also *rename* the directory. If a discrete profile is protecting the directory, the DIRECTRY profile will be renamed. If a generic profile is protecting the directory, the profile is not deleted, but an audit record may be written to record the directory being renamed.

Using the ICHDIRMV EXEC

The ICHDIRMV EXEC moves or renames one of your directory structures. This EXEC provides the same function as the CMS RENAME DIRECTORY and RELOCATE DIRECTORY commands. The syntax of ICHDIRMV is:

ichdirmv *dirid1 dirid2*

where:

dirid1 Is the name of the parent directory (top node) of the directory structure that is to be moved or renamed.

dirid2 Is the name of the new parent directory.

ICHDIRMV does the following:

1. Examines the structure of subdirectories and files under *dirid1*
2. Creates a similar directory structure under *dirid2*
3. Relocates all files in *dirid1* and its subdirectories with all their files into the new structure
4. Deletes *dirid1* and its directory structure
5. Moves corresponding RACF profiles from the old directory structure to the new directory structure.

The requirements for running ICHDIRMV for your own directory are:

- You must have at least one R/W disk or directory available in the search order that is not in the directory structure being moved.

- RACF must be active as the external security manager (ESM) for SFS, providing protection for SFS files and directories.
- Top-level generic profiles should exist for the directory structure being manipulated (directory profile FP:USER1.** and file profile * * FP:USER1.**, for example).

In addition, the requirements for running ICHDIRMV to move another user's directory include being authorized to issue the commands in Table 27 for the other user's directories:

Table 27. Authorization required to move another user's directory

Command	RACF Authority Required
CREATE LOCK	UPDATE authority to all subdirectories being moved
CREATE DIRECTORY	UPDATE authority to parent directories and ALTER to new directory names
RELOCATE file	If discrete RACF file profiles exist, you need group- or system-SPECIAL authority to rename them
RAC ADDDIR and RAC ADDFILE	Group- or system-SPECIAL authority
RAC DELDIR and RAC DELFILE	ALTER authority to profiles or group- or system-SPECIAL authority
DIRATTR (only issued if you are moving at least one DIRCONTROL directory)	SFSCMD authority to issue the DIRATTR command (or SFS file pool administrator authority if RACF is not protecting SFS administrator commands)
ERASE directory	ALTER authority to each subdirectory being moved

RACF Protection May Change

When ICHDIRMV is run, only RACF profiles within the directory structure being moved are moved to the new structure. There may be profiles in the old structure that protect more than just the files and directories in the old structure. *As a result, the RACF protection of some files or directories may be changed once the directory is moved to the new structure.* For example, suppose the following directories exist:

Table 28. RACF protection preceding ICHDIRMV command

SFS directories	RACF directory profile protecting the directory
FP:USER1.	FP:USER1.**
FP:USER1.DIR1	FP:USER1.**
FP:USER1.DIR1.SUB1	FP:USER1.**
FP:USER1.DIR2	FP:USER1.DIR2.**

After the command **ICHDIRMV FP:USER1.DIR1 FP:USER1.DIR2** is issued to relocate the DIR1 directory, the results are:

Table 29. RACF protection following ICHDIRMV command

SFS directories	RACF directory profile protecting the directory
FP:USER1.	FP:USER1.**
FP:USER1.DIR2	FP:USER1.DIR2.**
FP:USER1.DIR2.DIR1	FP:USER1.DIR2.** (was FP:USER1.**)
FP:USER1.DIR2.DIR1.SUB1	FP:USER1.DIR2.** (was FP:USER1.**)

The protection of the files within these directories may also change, depending on which file profiles existed before the directory move.

Debugging Options

The ICHDIRMV EXEC provides two keywords for debugging purposes. The syntax is:

ichdirmv *dirid1 dirid2* [**show** [**nocmd**]]

where:

show
Displays each CMS or RACF command as it is run

nocmd
Displays each CMS or RACF command that would be run, but does not actually run the commands, so the directory move is *not* performed

Recovery Process for ICHDIRMV

As the ICHDIRMV EXEC runs, it creates a recovery or backout file called **ICHDIRMV $$BACK$$**. This backout file contains backout commands corresponding to each SFS or RACF command that completes successfully. For example, if a **CREATE LOCK** command is issued successfully, a **DELETE LOCK** command is written to the backout file.

If an error occurs during processing, the backout commands are run to restore the user's original environment. If the backout is performed successfully, the backout file is erased. If any backout command fails, the successful and failing backout commands are recorded in the backout file also. In this case, the user should correct the errors that caused the backout commands to fail and then *reissue ICHDIRMV with no parameters*. ICHDIRMV will attempt to reissue the backout commands to restore the user's original environment. Again, the backout file will be updated to reflect the current status of the backout commands.

If ICHDIRMV is interrupted by a system outage or the file pool server becomes unavailable, the backout file will contain the commands issued before the interruption occurred. When the situation is corrected, the user should reissue ICHDIRMV with no parameters, and ICHDIRMV will attempt to back out any commands that ran successfully before ICHDIRMV was interrupted.

In a case where ICHDIRMV is interrupted unexpectedly, or an error occurs when changes are being backed out, directories in the structure being moved may remain locked. When ICHDIRMV is reissued, recovery will be attempted, and the locks will be removed.

Figure 24 on page 271 shows a sample backout file. *This file should never be edited.* In this example:

- The directory names entered when ICHDIRMV was issued are shown as **DIRID1** and **DIRID2**.
- The commands prefixed by **BACKOUTCMD** correspond to each CMS or RACF command issued successfully before an error or interruption occurred.
- The commands prefixed by **BACKOUTGOOD** are backout commands that ran successfully during backout processing.
- The commands prefixed by **BACKOUTFAIL** are backout commands that failed during backout processing. The failing commands should be corrected before running ICHDIRMV again.

The SFSAUTOACCESS Option

The SFSAUTOACCESS option, a performance enhancement for SFS FILE and DIRECTRY protection in RACF, reduces the number of RACROUTE calls made from SFS to RACF. This option allows the RACROUTE REQUEST=AUTH interface running in the SFS file pool server to automatically grant access to a file or directory if a user is accessing his or her own SFS file or directory. When automatically granting access, the call to the RACF service machine is bypassed, resulting in substantial performance improvements.

For more information, see *RACF System Programmer's Guide.*

Security Label Considerations for SFS Files and Directories

When the SECLABEL class is active and a security label exists in the FILE or DIRECTRY profile protecting the resource, RACF performs SECLABEL authorization checking. The type of SECLABEL authorization required is based on the type of access being requested for the SFS file or directory.

If the user requests an operation that requires:

1. READ access to the FILE or DIRECTRY profile, a read-only authorization request is performed.
2. READ, UPDATE, CONTROL, or ALTER access to the FILE or DIRECTRY profile, a read/write authorization request is performed.

The outcome of the request depends on the SETROPTS options that are in effect. For more information, see “Security Label Authorization Checking” on page 315.

When the SECLABEL class is active and SETROPTS MLACTIVE(FAILURES) has been issued, profiles in the FILE and DIRECTRY classes must contain a SECLABEL. Authorization requests for SFS files and directories without SECLABELs assigned to them will fail.

```
**PLEASE MAKE NO CHANGES TO THIS FILE**
DIRID1: RESEARCH:USER1.CDC1
DIRID2: RESEARCH:USER1.CDC1NEW
BACKOUTCMD: DELETE LOCK RESEARCH:USER1.CDC1
BACKOUTCMD: DELETE LOCK RESEARCH:USER1.CDC1.SUB1
BACKOUTCMD: DELETE LOCK RESEARCH:USER1.CDC1.SUB2
BACKOUTCMD: RAC DELDIR RESEARCH:USER1.CDC1NEW.**
BACKOUTCMD: RAC DELFILE * * RESEARCH:USER1.CDC1NEW.**
BACKOUTCMD: ERASE RESEARCH:USER1.CDC1NEW
BACKOUTCMD: ERASE RESEARCH:USER1.CDC1NEW.SUB1
BACKOUTCMD: ERASE RESEARCH:USER1.CDC1NEW.SUB2
BACKOUTCMD: DIRATTR RESEARCH:USER1.CDC1 DIRCONTROL
BACKOUTCMD: DIRATTR RESEARCH:USER1.CDC1.SUB1 DIRCONTROL
BACKOUTCMD: DIRATTR RESEARCH:USER1.CDC1.SUB2 DIRCONTROL
BACKOUTCMD: RELOCATE * * RESEARCH:USER1.CDC1NEW TO RESEARCH:USER1.CDC1
BACKOUTCMD: RELOCATE * * RESEARCH:USER1.CDC1NEW.SUB1 TO RESEARCH:USER1.CDC1.SUB1
BACKOUTCMD: RELOCATE * * RESEARCH:USER1.CDC1NEW.SUB2 TO RESEARCH:USER1.CDC1.SUB2
*=== END of Commands ===*
*=== BACKOUT starting ===*
BACKOUTGOOD: RELOCATE * * RESEARCH:USER1.CDC1NEW.SUB2 TO RESEARCH:USER1.CDC1.SUB2
BACKOUTGOOD: RELOCATE * * RESEARCH:USER1.CDC1NEW.SUB1 TO RESEARCH:USER1.CDC1.SUB1
BACKOUTGOOD: RELOCATE * * RESEARCH:USER1.CDC1NEW TO RESEARCH:USER1.CDC1
BACKOUTFAIL: DIRATTR RESEARCH:USER1.CDC1.SUB2 DIRCONTROL
BACKOUTFAIL: DIRATTR RESEARCH:USER1.CDC1.SUB1 DIRCONTROL
BACKOUTFAIL: DIRATTR RESEARCH:USER1.CDC1 DIRCONTROL
BACKOUTGOOD: ERASE RESEARCH:USER1.CDC1NEW.SUB2
BACKOUTGOOD: ERASE RESEARCH:USER1.CDC1NEW.SUB1
BACKOUTGOOD: ERASE RESEARCH:USER1.CDC1NEW
BACKOUTGOOD: RAC DELFILE * * RESEARCH:USER1.CDC1NEW.**
BACKOUTGOOD: RAC DELDIR RESEARCH:USER1.CDC1NEW.**
BACKOUTGOOD: DELETE LOCK RESEARCH:USER1.CDC1.SUB2
BACKOUTGOOD: DELETE LOCK RESEARCH:USER1.CDC1.SUB1
BACKOUTGOOD: DELETE LOCK RESEARCH:USER1.CDC1
```

Figure 24. Sample ICHDIRMV $$BACK$$ File

Implementing RACF Protection for SFS Files and Directories

To protect SFS files and directories using RACF, you'll need to:

1. Migrate existing SFS authorities into RACF using ICHSFS
2. Plan your changes to the DMSESM PROFILE
3. Activate RACF as the SFS external security manager

Step 1: Migrate Existing SFS Authorities Into RACF Using ICHSFS

You can use the ICHSFS EXEC to migrate existing SFS authorities into RACF. This EXEC does the following:

- Extracts SFS access control information for user-specified file pools by file and directory.
- For each file pool, generates a batch file of appropriate RACF commands to add equivalent access control information to the RACF database. The batch file has a file name equal to the file pool being migrated, and a file type of SFSUT1.

- When requested by the user, executes the batch file of RACF commands.
- If files with file type SFSUT1 exist when ICHSFS is invoked, the EXEC asks whether the files should be executed or deleted. If a user chooses not to execute or delete a batch file for a file pool, and later specifies that same file pool to be migrated, the existing batch file is erased and recreated.

Note: RACF does not recognize lowercase characters in profile names. The ICHSFS EXEC places any file names with lowercase characters in a file. You can use generic characters (including %) to protect these files. For example, to protect the file:

```
OFSMAIL OFSLOGfl POOL1:USER1.DIR1
```

you could use any of these file profile names:

```
OFSMAIL OFSLOG*  POOL1:USER1.DIR1
OFSMAIL OFSLOG%% POOL1:USER1.DIR1
*       OFSLOG%% POOL1:USER1.DIR1
*       OFSLOG%% POOL1:USER1.**
```

The ICHSFS EXEC can be used in two ways:

- Interactively, to process one file pool at a time
- Automatically, to process file pools listed in an input file.

Using ICHSFS Interactively

To use ICHSFS interactively, do the following:

1. While in a CMS ready state, enter:

   ```
   ichsfs
   ```

   In response, the EXEC prompts for the file pool to be migrated.

2. Enter the file pool ID.

   The EXEC creates a batch file of RACF commands for the file pool specified, and asks whether the batch file should be executed.

3. Enter yes or no.

   If the response is yes, the EXEC executes the batch file and prompts for another file pool to be migrated. If the response is no, the EXEC saves the batch file on the first R/W disk or directory in your search order.

Using ICHSFS Automatically

To use ICHSFS automatically, do the following:

1. Create a CMS file that lists the IDs of the file pools to be migrated. List one ID per line in the file, starting in column 1.

2. While in a CMS ready state, enter the following command:

   ```
   ichsfs filename filetype [filemode]
   ```

   where:

   filename is the file name of the file you created.

   filetype is the file type of the file you created.

   filemode is the file mode of the file you created. The file mode is optional. If it is not specified, it defaults to the first R/W disk or directory in your search order.

ICHSFS processes each file pool ID listed in the input file, creates a batch file of RACF commands for each one, and places the batch files on the first R/W disk or directory in your search order. When all of the file pools have been processed, the EXEC asks, one at a time, whether each batch file should be executed.

3. You can choose to execute each batch file, all of them, or none of them.

How SFS Authorities are Translated into RACF Authorities

To give all enrolled SFS users full access to their own SFS files and directories, the following steps are taken in the ICHSFS utility.

1. For every enrolled user, the following commands are issued:

   ```
   ADDDIR dirid.** OWNER(userid) UACC(NONE)
   PERMDIR dirid.** RESET
   ADDFILE * * dirid.** OWNER(userid) UACC(NONE)
   PERMFILE * * dirid.** RESET
   ```

2. For every directory, the following commands are issued:

   ```
   ADDDIR dirid OWNER(userid) UACC(NONE)
   PERMDIR dirid RESET
   ADDFILE * * dirid OWNER(userid) UACC(NONE)
   PERMFILE * * dirid RESET
   ```

3. For each file in a file control directory which has existing authorities granted to other users, the following commands are issued:

   ```
   ADDFILE fn ft dirid OWNER(userid) UACC(NONE)
   PERMFILE fn ft dirid RESET
   ```

Also, the file pool administrator who is running the ICHSFS utility is *not* given any authorities to the files and directories being migrated.

Table 30 on page 274 describes how previously issued GRANT commands (except those for <PUBLIC>) are translated into equivalent RACF authorities. Where there is not a direct equivalent in RACF for an SFS authority, the table notes the differences.

Table 30. SFS Authorities Translated As RACF Authorities

Type	SFS Authority	Corresponding RACF Authority
File control directory	READ	PERMDIR dirid ID(grantee) ACCESS(READ)
File control directory	WRITE	PERMDIR dirid ID(grantee) ACCESS(UPDATE) PERMFILE * * dirid ID(grantee) ACCESS(ALTER) **Note:** This also gives the equivalent of NEWWRITE authority.
File control directory	NEWREAD	PERMFILE * * dirid ID(grantee) ACCESS(READ)
File control directory	NEWWRITE	PERMFILE * * dirid ID(grantee) ACCESS(ALTER)
File in a File control directory	READ	PERMFILE fn ft dirid ID(grantee) ACCESS(READ)
File in a File control directory	WRITE	PERMFILE fn ft dirid ID(grantee) ACCESS(UPDATE) **Note:** The grantee is not allowed to erase the file.
Directory control directory	DIRWRITE	PERMDIR dirid ID(grantee) ACCESS(UPDATE) PERMFILE * * dirid ID(grantee) ACCESS(ALTER)
Directory control directory	DIRREAD	PERMDIR dirid ID(grantee) ACCESS(READ) PERMFILE * * dirid ID(grantee) ACCESS(READ)

Table 31 describes how previously issued GRANT commands for <PUBLIC> are translated into equivalent RACF authorities. Where there is not a direct equivalent in RACF for an SFS authority, the table notes the differences.

Table 31. SFS <PUBLIC> Authorities Translated As RACF Authorities

Type	SFS <PUBLIC> Authority	Corresponding RACF authority
File control directory	READ	ALTDIR dirid UACC(READ)
File control directory	WRITE	ALTDIR dirid UACC(UPDATE) ALTFILE * * dirid UACC(ALTER) **Note:** This also gives the equivalent of NEWWRITE authority.
File control directory	NEWREAD	ALTFILE * * dirid UACC(READ)
File control directory	NEWWRITE	ALTFILE * * dirid UACC(ALTER)
File in a File control directory	READ	ALTFILE fn ft dirid UACC(READ)
File in a File control directory	WRITE	ALTFILE fn ft dirid UACC(UPDATE) **Note:** Users are not allowed to erase the file.
Directory control directory	DIRWRITE	ALTDIR dirid UACC(UPDATE) ALTFILE * * dirid UACC(ALTER)
Directory control directory	DIRREAD	ALTDIR dirid UACC(READ) ALTFILE * * dirid UACC(READ)

Step 2: Plan Your Changes to the DMSESM PROFILE

The RACF protection of SFS files and directories is dependent on ESM calls from an SFS file pool server, and the DMSESM PROFILE defines which calls are made to RACF. The parameters of concern for SFS file and directory protection are the A*nnnn* parameter and the last two characters of the B*nnnn* parameter in Record 2.

See “Setting Up DMSESM PROFILE for RACF SFS Protection” on page 277 for more information.

Step 3: Activate RACF as the SFS External Security Manager

If you are activating RACF protection for the first time on your file pool server, follow the instructions in “Activating RACF as the SFS External Security Manager” on page 280.

If you are already using RACF to protect SFS administrator and operator commands or BFS files in your file pool, you must update the DMSESM PROFILE with the changes you planned in step 2, and restart your file pool server.

Controlling the Use of SFS Administrator and Operator Commands

You can activate the SFSCMD class to control who can issue SFS administrator and operator commands. Using SFSCMD profiles, users can provide protection for specific commands, and, through the use of generic profiles, for sets of commands.

The SFSCMD class is also used to restrict the use of the general user commands for RENAME DIRECTORY and RELOCATE DIRECTORY. See “Restrictions for SFS Directory RENAME and RELOCATE” on page 266 for more information.

When RACF is called to authorize an SFS administrator command, this replaces the SFS checking for file pool administrator authority, as specified in the DMSPARMS file or with the GRANT ADMIN command. RACF will authorize or deny use of the command by checking the command issuer's access to the SFSCMD profile protecting the command.

When RACF is protecting SFS operator commands, you must still enter the commands from the file pool server's console or from its secondary user console. Access to issue the commands can be further limited using SFSCMD profiles. RACF will authorize or deny use of the operator command by checking the command issuer's access to the SFSCMD profile protecting the command.

When SFS calls RACF to authorize an administrator or operator command and there is no profile to protect that command, RACF will fail the command.

To control which users can issue SFS administrator and operator commands, take the following steps:

1. Plan your changes to the DMSESM profile.

   The RACF protection of SFS administrator and SFS operator commands is dependent on ESM calls from an SFS file pool server using the RACROUTE interface. Using the DMSESM PROFILE as described in “Setting Up DMSESM PROFILE for RACF SFS Protection” on page 277, an installation can specify that RACF be called for all, some, or none of the SFS administrator and operator commands. The parameters of concern in the DMSESM PROFILE are the B*nnnn* and C*nn* parameters in Record 2.

2. If you plan to use generic profiles in the SFSCMD class, activate generic profiles:

   ```
   SETROPTS GENERIC(SFSCMD)
   ```

   IBM also recommends that you create a “top” generic profile in the SFSCMD class to protect those SFS administrator and operator commands that are not protected by a more specific profile. To do so, enter:

   ```
   RDEFINE SFSCMD ** UACC(NONE)
   ```

3. Create appropriate profiles in the SFSCMD class:

   ```
   RDEFINE  SFSCMD  profile-name  UACC(NONE)
   ```

   Specifying UACC(NONE) is recommended if most users on the system will *not* be able to issue SFS administrator and operator commands. The resource names are determined by VM, and are documented in *z/VM CMS File Pool Planning, Administration, and Operation.*

   For example, the SFS administrator command

   ```
   QUERY FILEPOOL REPORT (CATALOG
   ```

   is translated into the resource name

   ```
   filepool.QUERY.FILEPOOL.CATALOG
   ```

   in the SFSCMD class. This resource name could be protected by a RACF profile in the SFSCMD class named

   ```
   *.QUERY.FILEPOOL.CATALOG
   ```

   For example, the SFS operator command

   ```
   AUDIT ON ALL
   ```

   is translated into the resource name

   ```
   filepool.AUDIT.ON.ALL
   ```

   in the SFSCMD class. This resource name could be protected by a RACF profile in the SFSCMD class named

   ```
   *.AUDIT.ON.ALL
   ```

4. Use the PERMIT command to allow appropriate users and groups access to the profile by giving them at least UPDATE access:

   ```
   PERMIT profile-name CLASS(SFSCMD)  ID(userid|group)
          ACCESS(UPDATE)
   ```

   For example:

   ```
   PERMIT *.AUDIT.ON.ALL CLASS(SFSCMD) ID(LAURIE) ACCESS(UPDATE)
   ```

5. When you are ready to use the protection defined in SFSCMD profiles, activate the SFSCMD class:

   ```
   SETROPTS CLASSACT(SFSCMD)
   ```

6. For performance reasons, you should consider requesting SETROPTS RACLIST processing for the SFSCMD class:

   ```
   SETROPTS RACLIST(SFSCMD)
   ```

7. Activate RACF as the SFS external security manager.

If you are activating RACF protection for the first time on your file pool server, follow the instructions in “Activating RACF as the SFS External Security Manager” on page 280.

If you are already using RACF to protect SFS files and directories or BFS files in your file pool, you must update the DMSESM PROFILE with the changes you planned in step 1 above, and restart your file pool server.

Security Label Considerations for the SFSCMD class

When the SECLABEL class is active and a security label exists in the SFSCMD profile protecting the command, RACF performs SECLABEL authorization checking. A read/write authorization request is performed, and the outcome of the request depends on the SETROPTS options that are in effect. For more information, see “Security Label Authorization Checking” on page 315.

Setting Up DMSESM PROFILE for RACF SFS Protection

The file DMSESM PROFILE tells the SFS file pool server which types of authorization checking will be routed to an external security manager (ESM). This section describes what this file should look like when RACF is the designated ESM for SFS.

Details about each parameter specified in DMSESM PROFILE can be found in *z/VM CMS File Pool Planning, Administration, and Operation.*

The file ICHSFSPF DMSESM is shipped with RACF. It is a sample file for DMSESM PROFILE. Its contents determine which SFS authorization calls will be routed to RACF. It is illustrated in Figure 25. It may be modified as described below based on your installation's needs.

```
P1 DMSSECIT
A1012 DMSUAUTH B1012 DMSAAUTH C11 DMSOAUTH D1 DMSSECIT E1 DMSPERM
A002301 A002401 A002701 A002801
```

Figure 25. RACF-Supplied ICHSFSPF DMSESM

Record 1: Initialization and Termination Routine

The first line of the DMSESM PROFILE contains the name of the ESM initialization and termination routine. The line supplied with RACF specifies:

```
P1 DMSSECIT
```

where:

P1 Specifies that if RACF defers a request back to SFS (return code 4 from a RACROUTE REQUEST=AUTH, for example), REQUEST=AUTH, for example), SFS treats it as a rejected authorization. This value must not be changed.

DMSSECIT
Is the name of the routine called to initialize and terminate RACF processing for SFS. It must not be changed.

Record 2: Types of Calls to Be Reviewed

The second line of the DMSESM PROFILE identifies the types of authorization calls SFS should call RACF for. Five types of calls can be directed to RACF:

- SFS object authorization check
- SFS command authorization check
- SFS operator command authorization check
- ESM program check
- BFS object security

The line supplied with RACF specifies:

```
A1012 DMSUAUTH B1012 DMSAAUTH C11 DMSOAUTH D1 DMSSECIT E1 DMSPERM
```

where:

A1012

Specifies that RACF:

1. Checks new file, external object, and directory names as they are created
2. Does not check alias names as they are created, changed, or deleted
3. Is called by SFS for all object authorization checking
4. Is called by SFS at commit and rollback time only for specific file pool requests, identified by tokens in the third and following lines of the profile

To use SFSCMD protection with no file and directory protection, change this value to A0000. If you do this, you must also remove the A*nnnnnn* tokens on Record 3.

DMSUAUTH

Is the name of the routine called for object authorization checks. It must not be changed.

B1012

Specifies that:

1. RACF is called for SFS command authorization
2. RACF is not called at commit and rollback time for SFS command authorization
3. Renaming an SFS directory requires SFS command authorization
4. Relocating an SFS directory requires SFS command authorization

This value may be changed to B0012. This is the same as B1012 except RACF is not called for SFS command authorization. This would be used if you did not want to use SFSCMD profiles to protect any SFS administrator commands.

This value may also be changed to B2012 to specify that SFS calls RACF only for specific file pool requests, identified by tokens in the third and following lines of the profile. For example, if the third line contained B003210, SFS would call RACF for SFSCMD authorization when a delete storage request (file pool request code X'32') was made. See *z/VM CMS File Pool Planning, Administration, and Operation* for more information on using this value.

If you specify token A0000 to use SFSCMD protection with no file and directory protection, you can change this token to B1000 or B2000.

DMSAAUTH
Is the name of the routine called for command authorization checks. It must not be changed.

C11
Specifies that:

1. RACF is called for SFS operator command authorization
2. SFS will accept RACF denial of authorization for an operator command

This value may be changed to C01. This is the same as C11 except RACF is not called for SFS operator command authorization. This would be used if you did not want to use SFSCMD profiles to protect any SFS operator commands.

This value may also be changed to C21 to specify that SFS calls RACF only for specific file pool requests, identified by tokens in the third and following lines of the profile. For example, if the third line contained C000910, SFS would call RACF for SFSCMD authorization when a STOP command (special file pool request code `X'09'`) was issued. See *z/VM CMS File Pool Planning, Administration, and Operation* for more information on using this value.

DMSOAUTH
Is the name of the routine called for operator command authorization checks. It must not be changed.

D1 Specifies that SFS should call the specified routine if a program check occurs in RACF code. It must not be changed.

DMSSECIT
Is the name of the routine called to handle a program check in RACF code. It must not be changed.

E1 Specifies that RACF should be called for permission checking for BFS objects. If RACF is not being used to protect objects in BFS, this may be changed to **E0**.

See “Setting up Support for OpenExtensions” on page 238 for more information.

DMSPERM
Is the name of the routine called to perform permission checking for BFS objects. It must not be changed.

Record 3: Specific File Pool Requests To Be Reviewed

The third and following lines of the DMSESM PROFILE contain tokens identifying specific file pool requests that RACF should review. Each token represents a specific file pool request on a specific type of authorization call. The line supplied with RACF specifies:

```
A002301 A002401 A002701 A002801
```

where:

A002301
Specifies that RACF is called by SFS for an object authorization check for a *delete request* call at commit and rollback.

A002401
Specifies that RACF is called by SFS for an object authorization check for a *delete directory* call at commit and rollback.

A002701
Specifies that RACF is called by SFS for an object authorization check for a *relocate* call at commit and rollback.

A002801
Specifies that RACF is called by SFS for an object authorization check for a *rename* call at commit and rollback.

These four values must not be changed. However, if A0000 is specified in Record 2, you can delete these four values. In addition, you can add more B*nnnnnn* and C*nnnnnn* tokens to Record 3.

Activating RACF as the SFS External Security Manager

1. Authorize the file pool server to use the RACROUTE interface.

   Because SFS calls RACF using the RACROUTE interface, each file pool server which will be calling RACF must be authorized to use the RACROUTE interface. See *External Security Interface (RACROUTE) Macro Reference for MVS and VM* for instructions on doing this authorization using the ICHCONN profile in the FACILITY class. The access required for the ICHCONN profile is UPDATE authority.

   The file pool server also must have access to the RACROUTE interface code, which is typically placed on the Y-disk (MAINT's 19E) during RACF installation. The main module for the RACROUTE interface is the RPIUCMS MODULE.

   When RACF is specified as the external security manager for SFS, the number of calls to the RACF service machine increases dramatically. For this reason, IBM recommends that at least one RACF service machine be dedicated to process RACROUTE requests from the SFS service machine. If you have more than one SFS server, you can have them all send RACROUTE requests to one RACF service machine, or assign groups of them to different RACF service machines. This can be done using multiple RACF service machines, as described in *RACF System Programmer's Guide.*

2. Consider using the SFSAUTOACCESS option.

   If you are using RACF to protect SFS files and directories, and you want to use the SFSAUTOACCESS option, place a copy of the RACF SERVMACH file specifying the appropriate parameters on the file pool server's A-disk.

   See “The SFSAUTOACCESS Option” on page 270 and *RACF System Programmer's Guide* for more information.

3. Specify ESECURITY as a start-up parameter in the DMSPARMS file. This file usually resides on the A-disk of each SFS file pool server and is named *server-id* DMSPARMS, where *server-id* is the user ID of the file pool server. See *z/VM CMS File Pool Planning, Administration, and Operation* for more information.

4. Put DMSESM PROFILE on the file pool server's A-disk.

   You should have already planned the contents of the DMSESM PROFILE that is appropriate for your installation. Place this copy on the file pool server's A-disk.

5. Restart your file pool server.

   When you restart your file pool server, the RACROUTE interface will be initialized. The following messages are typically issued to the file pool server's console if initialization was properly completed:

```
DMS5SB2029I Initialization begins for external security
DMS5SB2029I Initialization begins for external security routine DMSSECIT
RPICMS016I USER/RACF VM Racroute communication path is established.
DMS5SB2029I Initialization ends for external security routine DMSSECIT
```

SFS Administration with RACF

This section will discuss some RACF administration topics associated with SFS.

Using SPECIAL and OPERATIONS Authorities

When SFS is active without using an external security manager, an SFS administrator has full access over all resources in a file pool. For example, the administrator can read from or write to any user file in the file pool and also grant and revoke authorities to users' files and directories. With RACF, there is no direct equivalent to this type of authority for a single file pool. However, the RACF SPECIAL and OPERATIONS attributes offer similar capabilities.

The RACF SPECIAL attribute allows you to update any profile in the RACF database. This includes FILE and DIRECTRY profiles, but also all other profiles. With the SPECIAL attribute, you can update the access list in any profile to authorize users to access resources. You can also update other fields in RACF profiles such as UACC, NOTIFY, and LEVEL. For more information on the RACF SPECIAL attribute, see “SPECIAL Attribute” on page 57.

You can also use group-SPECIAL authority to limit the scope of SPECIAL authority. For more information, see “Scope of Authority for the group-SPECIAL, group-AUDITOR, and group-OPERATIONS Users” on page 60.

Note: To create a FILE or DIRECTRY profile containing a second qualifier that is different from your own user ID, you must have either the SPECIAL attribute or group-SPECIAL authority.

The RACF OPERATIONS authority allows you to read and write from *any* SFS file or directory in *any* file pool. With the OPERATIONS attribute, you can also access resources in other classes, such as minidisks. For more information on the RACF OPERATIONS attribute, see “OPERATIONS Attribute” on page 58.

You can also use group-OPERATIONS authority to limit the scope of OPERATIONS authority. For more information, see “Scope of Authority for the group-SPECIAL, group-AUDITOR, and group-OPERATIONS Users” on page 60.

Disallowed Commands

Table 32 describes commands that are not allowed when SFS is running with RACF protection active. Along with each item is the RACF replacement for it and where to find more information.

Table 32. Disallowed Commands

Disallowed Commands:	Replaced With RACF:	Where To Find More Information:
End-user commands: GRANT REVOKE	RAC PERMDIR command RAC PERMFILE command	See “End-User Interaction with SFS and RACF.”
SFS administrator commands: ENROLL ADMINISTRATOR DELETE ADMINISTRATOR	SFSCMD authorities RACF SPECIAL authority RACF OPERATIONS authority	See “Controlling the Use of SFS Administrator and Operator Commands” on page 275 and “Using SPECIAL and OPERATIONS Authorities” on page 281.
SFS operator commands: GRANT ADMIN REVOKE ADMIN	SFSCMD authorities RACF SPECIAL authority RACF OPERATIONS authority	See “Controlling the Use of SFS Administrator and Operator Commands” on page 275 and “Using SPECIAL and OPERATIONS Authorities” on page 281.

Adding an SFS User

When you enroll a new user into an SFS file pool, you should also create "top" generic profiles and make the user the owner of the profiles. For example, assuming that user JOSE's file pool ID is POOL1, you would do the following:

```
RAC ADDFILE * * POOL1:JOSE.** OWNER(JOSE) UACC(NONE)
RAC ADDDIR POOL1:JOSE.** OWNER(JOSE) UACC(NONE)
```

Notes:

1. All users can create and modify their own FILE and DIRECTRY profiles. Class authority, or CLAUTH, does not apply to the FILE and DIRECTRY classes.
2. Users always have access to their own files and directories. This is determined by the user ID qualifier of the FILE or DIRECTRY profile name.

End-User Interaction with SFS and RACF

This section discusses the RACF commands that replace the functions of several CMS commands.

Replacements for the GRANT and REVOKE Commands

When RACF protection is in effect for SFS, you cannot use the CMS GRANT and REVOKE commands, Instead, use the RACF PERMDIR and PERMFILE commands to authorize another user to access your SFS files and directories. Table 33 on page 283 describes how SFS authorities used on the GRANT command can be matched to equivalent RACF authorities. Where there is not a direct equivalent in RACF for an SFS authority, the table notes the differences.

Table 33. SFS Authorities Translated As RACF Authorities: For End Users

Type of Resource	SFS Authority	Corresponding RACF authority
File control directory	READ	RAC PERMDIR dirid ID(grantee) ACCESS(READ)
File control directory	WRITE	RAC PERMDIR dirid ID(grantee) ACCESS(UPDATE) RAC PERMFILE * * dirid ID(grantee) ACCESS(ALTER) **Note:** This also gives the equivalent of NEWWRITE authority.
File control directory	NEWREAD	RAC PERMFILE * * dirid ID(grantee) ACCESS(READ)
File control directory	NEWWRITE	RAC PERMFILE * * dirid ID(grantee) ACCESS(ALTER)
File in a File control directory	READ	RAC PERMFILE fn ft dirid ID(grantee) ACCESS(READ)
File in a File control directory	WRITE	RAC PERMFILE fn ft dirid ID(grantee) ACCESS(UPDATE) **Note:** The grantee is not allowed to erase the file.
Directory control directory	DIRWRITE	RAC PERMDIR dirid ID(grantee) ACCESS(UPDATE) RAC PERMFILE * * dirid ID(grantee) ACCESS(ALTER)
Directory control directory	DIRREAD	RAC PERMDIR dirid ID(grantee) ACCESS(READ) RAC PERMFILE * * dirid ID(grantee) ACCESS(READ)

Table 34 describes how you can do the equivalent of granting <PUBLIC> authority with the RACF UACC keyword. Where there is not a direct equivalent in RACF for an SFS authority, the table notes the differences.

Table 34. SFS <PUBLIC> Authorities Translated As RACF Authorities: For End Users

Type of Resource	SFS <PUBLIC> Authority	Corresponding RACF authority
File control directory	READ	RAC ALTDIR dirid UACC(READ)
File control directory	WRITE	RAC ALTDIR dirid UACC(UPDATE) RAC ALTFILE * * dirid UACC(ALTER) **Note:** This also gives the equivalent of NEWWRITE authority.
File control directory	NEWREAD	RAC ALTFILE * * dirid UACC(READ)
File control directory	NEWWRITE	RAC ALTFILE * * dirid UACC(ALTER)
File in a File control directory	READ	RAC ALTFILE fn ft dirid UACC(READ)
File in a File control directory	WRITE	RAC ALTFILE fn ft dirid UACC(UPDATE) **Note:** Users are not allowed to erase the file.
Directory control directory	DIRWRITE	RAC ALTDIR dirid UACC(UPDATE) RAC ALTFILE * * dirid UACC(ALTER)
Directory control directory	DIRREAD	RAC ALTDIR dirid UACC(READ) RAC ALTFILE * * dirid UACC(READ)

Replacement for the QUERY AUTHORITY Command

The CMS QUERY AUTHORITY command displays the authorities for an SFS file or directory. When RACF is protecting SFS files and directories, the QUERY AUTHORITY command will display a "P" to indicate a file or directory is protected by RACF. Instead of the QUERY AUTHORITY command, use the RACF LFILE and LDIRECT commands to display the authorities for SFS files and directories.

The LFILE and LDIRECT commands list the RACF profile protecting a resource. For example, to list the profile and authorities for ANDREW's top directory in file pool POOL1, issue:

```
RAC LDIRECT POOL1:ANDREW. AUTH
```

See *RACF General User's Guide* and *RACF Command Language Reference* for details on using the LFILE and LDIRECT commands.

Replacement for the RENAME and RELOCATE DIRECTORY Commands

The RENAME DIRECTORY and RELOCATE DIRECTORY commands should not be used when RACF is protecting SFS files and directories. Users should use the ICHDIRMV EXEC instead. See “Restrictions for SFS Directory RENAME and RELOCATE” on page 266 and “Using the ICHDIRMV EXEC” on page 267 for more information.

Chapter 16. Operating Considerations

As the security administrator, you should be familiar with the operating considerations discussed in this chapter.

Coordinating Profile Updates

You should plan to update profiles so they remain consistent with other profiles on the database while making sure the updating process does not interfere with other jobs running in the system.

Each individual operation performed by RACF serializes on a RACF database, but a command or function may perform multiple operations on multiple profiles. For example, the CONNECT command changes both the user profile and the group profile. If two or more RACF commands or functions are executing at the same time and are making contradictory updates, their operations might be interleaved and, therefore, cause the information in the RACF database to become incomplete or invalid.

Note: If a user is logged on, and you update the user's attributes in the RACF database using ALTUSER or CONNECT, some changes may not take effect until the next time the user enters the system. Some of these changes that are delayed until the user logs on again are RACF SPECIAL, RACF OPERATIONS, RACF AUDITOR, and the list of connected groups examined by RACROUTE REQUEST=FASTAUTH. However, a LISTUSER or LISTGRP command issued immediately after the change shows the new values.

In the following example, the system administrator inadvertently creates a situation where a profile exists, but it does not have an owner.

Example:

1. The system administrator deletes a user who is logged on.
2. That user, while logged on, defines a profile for a resource.

This creates an *ownerless* profile.

To prevent the creation of ownerless profiles, do not delete a user who is logged on. Instead, prevent the user from logging on by revoking that user with the ALTUSER command before deleting the user profile. If necessary, have the operator force the user off the system, then revoke the user, and delete the user profile when you have completed your preparations. See “Summary of Steps for Deleting Users on VM” on page 75.

Getting Started After RACF Is Installed

During RACF installation, a basic set of profiles is created in the RACF database:

Group	Superior Group	Owner	Connected Users (Group Authority)
SYS1	–	IBMUSER	IBMUSER (JOIN)
VSAMDSET	SYS1	IBMUSER	IBMUSER (JOIN)
SYSCTLG	SYS1	IBMUSER	IBMUSER (JOIN)

And there is only one user:

User	Default Group (Group Authority)	Attributes	Connected Groups (Group Authority)
IBMUSER	SYS1 (JOIN)	SPECIAL and OPERATIONS	SYSCTLG (JOIN) VSAMDSET (JOIN)

Logging on as IBMUSER and Check Initial Conditions

IBMUSER is the first user ID that the security administrator can use. This user ID has the system-SPECIAL attribute, which allows IBMUSER to issue most of the RACF commands (except those reserved for users with the AUDITOR attribute) and the system-OPERATIONS attribute, which allows IBMUSER to access many RACF-protected resources.

When entering the system for the first time with the IBMUSER user ID, you must change the initial password, SYS1, to a new password. A new password prevents any other user from entering the system as IBMUSER.

Log on as IBMUSER:

```
LOGON IBMUSER
```

After entering IBMUSER's old password (SYS1) and defining a new password, enter a RACF command session:

```
RACF
```

Next, list the system-wide RACF options that are in effect:

```
SETROPTS LIST
```

All options will not be displayed at this point because IBMUSER does not have the AUDITOR attribute. If you want to see the status of those options, grant IBMUSER this attribute and have him log off and log on again. Issue SETROPTS LIST to see a display of these options.

Read through the list of parameters to familiarize yourself with what options are in effect and what they mean.

Attention

The option for the TERMINAL resource class should be specified as READ. Do not change it to NONE unless you have defined your terminals to RACF and authorized the appropriate users and groups to access them. If you specify TERMINAL(NONE) without first defining your terminals to RACF, you will not be able to access your terminals and, consequently, you will be locked out of your system.

You should also check which VM events are controlled or audited by issuing the following command:

```
SETEVENT LIST
```

Defining Administrator User IDs for Your Own Use

Define a new user (for example, RACFADM) to RACF for your own use. This user should have at least the SPECIAL and OPERATIONS attributes. If you will also act as the system-wide auditor, you should also give this user ID the AUDITOR attribute. Do one of the following:

```
ADDUSER RACFADM SPECIAL OPERATIONS
```

or:

```
ADDUSER RACFADM SPECIAL OPERATIONS AUDITOR
```

Note: Make sure the new administrator has access to the minidisk that contains the RPIDIRCT SYSUT1 file. For example:

```
RDEFINE  VMMDISK  RACFVM.301
PERMIT  RACFVM.301  CLASS(VMMDISK)  ID(RACFADM)  ACCESS(UPDATE)
```

Then, exit the RACF command session:

```
END
```

Logging On as RACFADM, Checking Groups and Users, and Revoking IBMUSER

Log on as RACFADM and use the default password, SYS1 in this case (IBMUSER's default group).

First, list all users to ensure that only RACFADM and IBMUSER are defined to RACF, and that they have the proper attributes.

At this point, you will receive a message stating that your password expired. Immediately change the password, SYS1, to a new password.

```
LISTUSER *
```

Then, list all the groups defined to RACF:

```
LISTGRP *
```

Connect RACFADM to them and make RACFADM the owner of the groups, for example:

```
CONNECT RACFADM GROUP(SYS1)  AUTH(JOIN)
CONNECT RACFADM GROUP(SYSCTLG)  AUTH(JOIN)
CONNECT RACFADM GROUP(VSAMDSET) AUTH(JOIN)
ALTGROUP SYS1     OWNER(RACFADM)
ALTGROUP SYSCTLG  OWNER(RACFADM)
ALTGROUP VSAMDSET OWNER(RACFADM)
```

To verify that the changes took place, issue the following commands:

```
LISTGRP SYS1
LISTGRP SYSCTLG
LISTGRP VSAMDSET
```

Then, revoke IBMUSER so that another user cannot make use of the IBMUSER user ID:

```
ALTUSER IBMUSER REVOKE
```

Note: You cannot delete the IBMUSER user profile.

Define another user to RACF (user ID RACFAD2), who will act as your assistant. Make the new user's default group SYS1, and give this assistant the SPECIAL and OPERATIONS user attributes.

```
ADDUSER RACFAD2 DFLTGRP(SYS1) AUTH(JOIN) SPECIAL OPERATIONS
```

Setting RACF Options

Review Chapter 13, “Selecting RACF Options on VM (SETROPTS)” on page 215 for the RACF options you wish to set.

For information about:

- Selecting options with the SETROPTS command, see Chapter 13, “Selecting RACF Options on VM (SETROPTS)” on page 215
- Encrypting RACF user passwords, see “Encryption of RACF User Passwords” on page 208.

Defining the Groups Needed for the First Users

At this point you should consider creating the groups that you will need. You proceed to add four groups: three (GROUP1, GROUP2, and GROUP3) are departmental groups, with GROUP2 and GROUP3 owned by GROUP1 so that certain authorities can be propagated. The fourth group (DATAMGT) has minidisk maintenance responsibility.

```
ADDGROUP (GROUP1 DATAMGT)
ADDGROUP (GROUP2 GROUP3) OWNER(GROUP1) SUPGROUP(GROUP1)
```

Defining a System-Wide Auditor

Define a user who will have system-wide auditing responsibilities and privileges.

```
ADDUSER AUDCCC AUDITOR
```

Defining Users and Groups

You now add a user (D03DIK) to GROUP3.

```
ADDUSER D03DIK OWNER(GROUP3) DFLTGRP(GROUP3)
```

The password is given by the security administrator, and this initial password is the name of the default group.

Note: For more information on defining users, see “Summary of Steps for Defining Users on VM” on page 74.

Defining Group Administrators, Group Auditor, and Data Manager

Define a group administrator with the group-SPECIAL attribute for each group. Only the administrator for GROUP1 has authority to define new users in that group. Each of the other administrators has authority over resources owned by his or her group, as well as resources owned by users who are owned by his or her group.

```
ADDUSER D01RHG DFLTGRP(GROUP1) CLAUTH(USER) DATA('GROUP1 ADM')
CONNECT D01RHG GROUP(GROUP1) AUTH(JOIN) SPECIAL
ADDUSER D02JMP DFLTGRP(GROUP2) DATA('GROUP2 ADM')
CONNECT D02JMP GROUP(GROUP2) SPECIAL
ADDUSER D03ABL DFLTGRP(GROUP3) DATA('GROUP3 ADM')
CONNECT D03ABL GROUP(GROUP3) SPECIAL
```

Define group-auditor for groups GROUP1, GROUP2, and GROUP3. Connect the user to GROUP1 and give the user the group-AUDITOR attribute. Because GROUP2 and GROUP3 are owned by GROUP1, the user has auditor authority over the resources and users belonging to those groups, as well as to GROUP1. The user does not have auditor authority in any other group.

```
ADDUSER D01GPB DFLTGRP(GROUP1) DATA('AUDITOR G1 G2 G3')
CONNECT D01GPB GROUP(GROUP1) AUDITOR
```

The administrator for the data management group, the data manager, is able to define VM minidisks to RACF in order to perform dump, restore, and data cleanup operations.

```
ADDUSER DMGJFS DFLTGRP(DATAMGT) AUTH(JOIN)
        CLAUTH(USER VMMDISK) DATA('DATA MGT ADM')
```

Because of their duties, data managers are connected to SYS1, allowing them to access resources with SYS1 in their access list. The data manager has the group-SPECIAL attribute in group SYS1.

```
CONNECT DMGJFS GROUP(SYS1) UACC(READ) SPECIAL
```

Exit the RACF command session:

```
END
```

At the end of the session, the defined group structure is:

Group	Superior Group	Owner	Connected Users (Group Authority)
SYS1	–	RACFADM	IBMUSER (JOIN) RACFADM (JOIN) RACFAD2 (JOIN) DMGJFS (CREATE)
VSAMDSET	SYS1	RACFADM	IBMUSER (JOIN) RACFADM (JOIN)
SYSCTLG	SYS1	RACFADM	IBMUSER (JOIN) RACFADM (JOIN)
GROUP1	SYS1	RACFADM	D01RHG (JOIN) D01GPB (USE)
GROUP2	SYS1	GROUP1	D02JMP (CREATE)
GROUP3	SYS1	GROUP1	D03ABL (CREATE)
DATAMGT	SYS1	RACFADM	DMGJFS (JOIN)

The defined users are:

User	Default Group (Group Authority)	Attributes	Connected Groups (Group Authority)
IBMUSER	SYS1 (JOIN)	SPECIAL, OPERATIONS, REVOKE	SYSCTLG (JOIN), VSAMDSET (JOIN)
RACFADM	SYS1 (JOIN)	SPECIAL, AUDITOR, OPERATIONS	SYSCTLG (JOIN), VSAMDSET (JOIN)
RACFAD2	SYS1 (JOIN)	SPECIAL, OPERATIONS	
DMGJFS	DATAMGT (JOIN), SYS1 (CREATE)	CLAUTH(USER VMMDISK), SPECIAL, OPERATIONS	SYS1 (CREATE)
D01RHG	GROUP1 (JOIN)	CLAUTH(USER), group-SPECIAL	
D02JMP	GROUP2 (USE)	group-SPECIAL	
D03ABL	GROUP3 (CREATE)	group-SPECIAL	
D01GPB	GROUP1 (CREATE)	group-AUDITOR	
D03DIK	GROUP3 (CREATE)		
AUDCCC	SYS1 (USE)	AUDITOR	

The RAC Command Processor

Users can use the RAC command processor to issue RACF commands. The RAC command processor allows users to issue RACF commands without isolating themselves in a RACF command session.

Note: RAC cannot be issued from a RACF service machine.

For example, instead of the following sequence:

```
RACF
ADDUSER JOE
RDEFINE VMMDISK JOE.191 UACC(NONE)
END
```

a user could issue:

```
RAC ADDUSER JOE
RAC RDEFINE VMMDISK JOE.191 UACC(NONE)
```

This allows the user to issue CMS or CP commands between the RACF commands.

Also, the RAC command processor allows your installation to tailor the RACF commands so that users need not follow the syntax of the RACF commands as shipped by IBM. (For more information, see *RACF System Programmer's Guide*.)

Using DSMON

The data security monitor (DSMON) produces a set of reports that provide information about the current status of the data security environment at your installation. These reports can help you to (1) check the initial steps you took to establish system security, and (2) make additional security checks periodically.

On VM systems, the reports DSMON can produce are:

- System report
- Selected user attribute reports
- Selected data sets report
- RACF exits report
- Class descriptor table report
- Global access table report
- Group tree report

A short description of each report follows. For more information on these reports and how to invoke the data security monitor, see *RACF Auditor's Guide*.

System Report

This report contains information such as the identification and model of the processor complex. This report also specifies 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.

Selected User Attribute Reports

The selected user attribute report lists all RACF users with the SPECIAL, OPERATIONS, AUDITOR, or REVOKE attributes and specifies whether they possess these attributes on a system-wide (user) or group level. You can use this report to verify that only those users who need to be authorized to perform certain functions have been assigned the corresponding attribute.

The selected user attribute summary report shows the number of installation-defined users and totals for users with the SPECIAL, OPERATIONS, AUDITOR, and REVOKE attributes, at both the system and group level. You can use this report to verify that the number of users with each of these attributes, on either a system or group level, is the number that your installation wants. In particular, you should make sure that you have assigned the SPECIAL attribute (on a system level) to at least one user and the AUDITOR attribute (on a system level) to at least one user.

Selected Data Sets Report

This report lists the names of the primary and backup RACF database(s). You can use the selected data sets report to determine if your RACF data sets are protected by RACF.

RACF Exits Report

This report lists the names of all the installation-defined RACF exit routines and specifies the size of each exit routine module.

You can use the RACF exits report to verify that the only active exit routines are those that your installation has defined. The existence of any other exit routines might indicate a system security exposure, because RACF exit routines could be used to bypass RACF security checking. Similarly, if the length of an exit routine module differs from the length of the module when it was defined by your installation, the module might have unauthorized modifications.

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.

You can use the class descriptor table report to determine which classes (besides DATASET) are defined to RACF and active, and therefore can contain resources that RACF protects.

Global Access Table Report

This report lists, for each resource class in the global access table, all the entry names and their associated resource access authorities.

Because global access checking allows anyone to access the resource at the associated access authority, you should verify that each entry has an appropriate level of access authority.

Group Tree Report

This report lists, for each requested group, all its subgroups, all the subgroups' subgroups, and so on, as well as the owner of each group listed in the report, if the owner is not the superior group.

You can use the group tree report to examine the overall RACF group structure for your system. You can also determine the scope of the group for group-related user attributes (group-SPECIAL, group-OPERATIONS, and group-AUDITOR).

Controlling Access to RACF Passwords

Installation personnel should ensure that the security of RACF user logon passwords is not violated. It is possible for an operator to display password information when displaying real storage at a console. The installation should monitor the operator's activities to ensure passwords remain secure. In addition, you should restrict access to dumps that might contain password information.

Deactivating RACF

On VM, you can deactivate RACF as follows:

- You can deactivate the RACF database by using the RVARY command with the INACTIVE operand. (You must deactivate the RACF database, for example, if you want to perform maintenance on it.) In this situation, failsoft processing takes effect. Each time you request access to a resource, RACF, through failsoft processing, prompts the operator to validate your request. The operator then either grants or denies you access to the resource. (For more information on failsoft processing, see *RACF System Programmer's Guide* and the following section of this chapter.)

  Notes:

  1. You cannot log on while the RACF database is inactive.
  2. RACF does not automatically propagate the RVARY command to other MVS or VM systems sharing the database; this command must be issued to each system sharing the database.

     Note: If multiple service machines on a single VM system share the RACF database, the command must be propagated to each service machine. To coordinate the command across multiple service machines, refer to the RAC command information in *RACF Command Language Reference*.

- You can deactivate RACF itself by using the SETRACF command with the INACTIVE operand. (You might do this, for example, if you need to perform maintenance on RACF itself immediately after it is installed.) Deactivating RACF in this manner causes CP to handle all authorization requests as they were prior to installing RACF.

  Note: When you use the SETRACF command, it applies to all service machines on that VM system.

- You can also deactivate RACF itself by modifying the ICHSECOP module.

Attention

Notes:

1. Deactivating RACF by modifying ICHSECOP is not recommended for use on VM systems. It results in a situation where no users, including users who are already logged on, can access the system or any of its resources.
2. The RVARY INACTIVE command will cause a lockout if it is issued using RAC. Because RAC issues commands directly to the active RACF service machine, it would be unable to issue an RVARY ACTIVE command when RACF is inactive. Use of the RVARY command should be limited to a RACF command session using the RACF EXEC.

Failsoft Processing

During failsoft processing (when the RACF database is not active), RACF uses global access checking tables, or a supplied profile, if any of these are present, to process resource access checking requests. (Note that RACF does not perform generic profile checking because a generic profile might allow access to a resource that an existing discrete profile already protects; had that profile been retrieved, RACF would not have allowed access to the resource.)

RACF calls RACROUTE REQUEST=AUTH and RACROUTE REQUEST=DEFINE preprocessing installation exits during failsoft processing. (RACF does not call the postprocessing exits.) This action frees the installation to define its own version of failsoft processing. By defining its own version of failsoft processing, an installation can allow or deny access to a resource or permit normal failsoft processing to continue.

During failsoft processing the logging that your installation has specified continues as when RACF is active. In addition, RACF logs all accesses that the operator allows or denies.

If no global access checking tables are present, and no profile has been supplied, RACF calls the preprocessing installation exits. Failsoft processing then continues as follows:

- **RACROUTE REQUEST=AUTH:** RACF issues an operator intervention message to request permission to allow access to the resource.
- **RACROUTE REQUEST=DEFINE:** RACF issues an operator message to indicate that RACROUTE REQUEST=DEFINE has been issued and that the request is allowed.

  You can use the operator message or SMF log records at a later time to determine whether the specified resource is in the RACF database. If it is not, use the RDEFINE command to create a profile for the resource.

Note: Failsoft processing is not in effect when you deactivate RACF by using SETRACF INACTIVE or by modifying the ICHSECOP module.

Service by IBM Personnel

If IBM support personnel require access to the system for servicing, they must be defined to RACF if they need to access RACF-protected resources for servicing. Also, they need the appropriate access authority to these resources.

You can define user profiles for IBM support personnel with the REVOKE attribute set. Then an authorized installation user can set (and reset), as needed, the REVOKE attribute in the user profile to allow IBM support personnel to enter the system. (The REVOKE and RESUME operands of the ALTUSER or CONNECT command alter the REVOKE attribute. See *RACF Command Language Reference* for more information.)

Considerations for the RACF Database

As an alternative to maintaining all of your RACF profiles on one database, RACF allows you to have multiple (up to four on VM) RACF databases. The use of multiple RACF databases is recommended to reduce device contention and to reduce the number of resources made unavailable by the loss of one database or device. The optimum number of RACF databases for your installation depends on the extent to which you use RACF.

Note that the RACF databases should, themselves, be RACF-protected.

Protecting the RACF Database

The minidisks containing the RACF database should, themselves, be RACF-protected with minidisk profiles with UACC(NONE), NOWARNING, and ERASE specified. The NOTIFY user ID should be the RACF security administrator. System programmers who may need to use the BLKUPD command to repair the RACF database must have UPDATE authority to the RACF database.

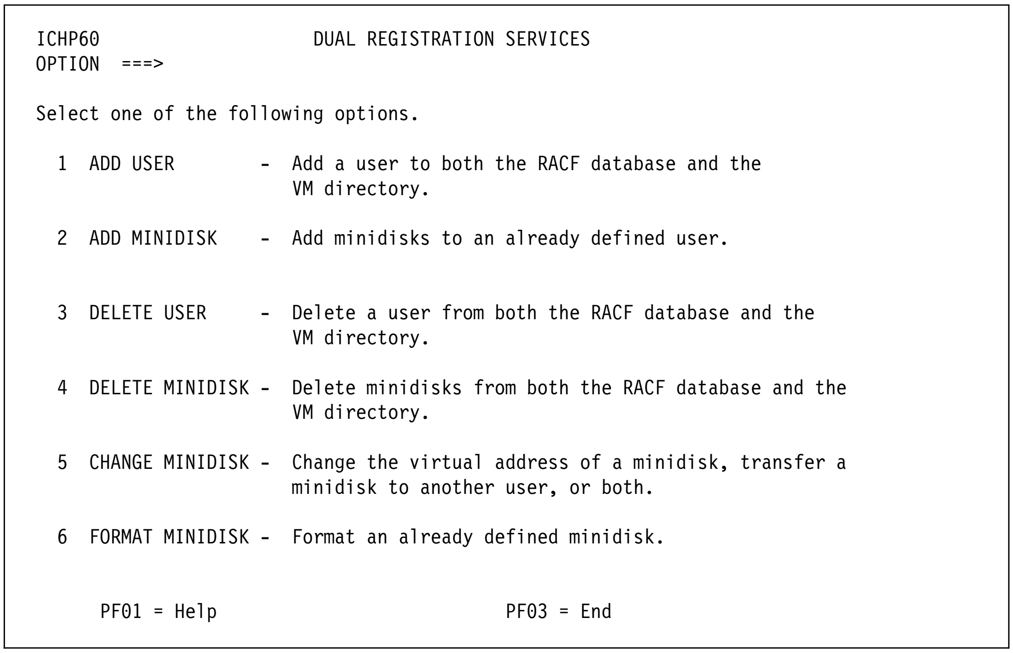
See *RACF System Programmer's Guide* for details on how to maintain RACF databases, switch to alternate RACF databases, coordinate profile updates, and make other considerations when dealing with shared or multiple RACF databases.

Number of Resident Data Blocks

The use of resident data blocks reduces the number of I/O requests made to the RACF database. The default number of resident data blocks on VM is 100. If you need to change the number of resident data blocks, see *RACF System Programmer's Guide* for details.

Using Dual Registration Panels

On VM systems, RACF provides dual registration panels so that you can add or delete users and minidisks in the RACF database and the VM directory at the same time. You can also use the dual registration panels to transfer a minidisk from one user to another. The dual registration menu, shown in Figure 26 on page 297, shows what you can do.



```
ICHP60                    DUAL REGISTRATION SERVICES
OPTION  ===>

Select one of the following options.

  1  ADD USER        -  Add a user to both the RACF database and the
                        VM directory.

  2  ADD MINIDISK    -  Add minidisks to an already defined user.


  3  DELETE USER     -  Delete a user from both the RACF database and the
                        VM directory.

  4  DELETE MINIDISK -  Delete minidisks from both the RACF database and the
                        VM directory.

  5  CHANGE MINIDISK -  Change the virtual address of a minidisk, transfer a
                        minidisk to another user, or both.

  6  FORMAT MINIDISK -  Format an already defined minidisk.


      PF01 = Help                        PF03 = End
```

Figure 26. Dual Registration Menu

To use dual registration panels, your installation must have ISPF Version 3 Release 2 or later and DirMaint Version 1 Release 5 or later.

When you install dual registration panels, you can specify default values for certain fields that appear on the ISPF dual registration panels by using the file DUALREG PROFILE. For example, you might want to protect all minidisks that you define through dual registration panels with a universal access authority (UACC) of NONE. You can specify a UACC of NONE for minidisks in DUALREG PROFILE, and this value will appear in the UACC field on all of the appropriate dual registration panels. (This is generally done during installation of RACF. For more information on installing RACF, see *RACF Program Directory*.)

When using the panels, users can override the default values that appear on the ISPF dual registration panels. For example, suppose you have specified a default UACC of NONE for minidisks in DUALREG PROFILE and you want to define a system minidisk, SYSDISK.191, and give it a UACC of READ to allow all users to read it. When you access the appropriate dual registration panel to define SYSDISK.191, a value of NONE will appear in the UACC field. You can change the UACC to READ and this value will become the UACC for SYSDISK.191. However, the default UACC of NONE for minidisks specified in DUALREG PROFILE remains unchanged and will appear on the appropriate dual registration panel the next time you define a minidisk.

In order to use the dual registration panels, all of the following must be true:

1. You must be running with DirMaint command level 140A. If you are running with command level 150A, you can change your default command level to 140A with the following command before using the dual registration panels:

   ```
   DIRM DEFAULTS CMDLEVEL 140A
   ```

2. You must be authorized to issue the following DirMaint commands: ADD, AMDISK, CHVADDR, DIRECT, DMDISK, PURGE, and TMDISK in command level 140A for *all* user IDs. By default, these DirMaint commands are in

command sets A and D. If running with ADD_COMMAND_PROCESSING=SHORT or with PURGE_COMMAND_PROCESSING=SHORT, the ADD and PURGE commands have probably been changed to command set S.

3. If you are adding user IDs with LINK statements in their directory entries, you also must be authorized for command set G for *all* users.

Note: The dual registration dialog does not support the use of the ACIGROUP control statement.

Using RACF 1.10 with Other VM Products

Attachable Media Manager/VM (AMMR)

Attachable Media Manager/VM is a licensed program that gives an installation the means of controlling access to attachable hardware devices such as lines, printers, tapes and removable disks. AMMR/VM provides commands that enable users to keep track of volumes they own, as well as control access to them and obtain or release volumes as required.

The task of AMMR is to enforce an installation's mandatory (or discretionary) access control policy on all objects in the library. AMMR/VM provides the following support:

- Uses the RACROUTE interface for validating, authorizing and auditing tape volume access. This support uses profiles in the TAPEVOL class.
- External security manager interface for verifying and auditing user authorization to issue commands as well as user authorization to access volumes, to ensure separation of privileged and general user commands. This support uses profiles in the FACILITY class.
- Provide standard tape label processing, including bypass label protection and no label processing. This support uses profiles in the FACILITY class.

For additional information on using AMMR, see:

- *AMMR/VM Installation and Operations*
- *AMMR/VM Command Language Reference*
- *VM/ESA C2/B1 Trusted Facility Manual for VM/ESA with RACF.*

Print Services Facility/VM (PSF)

Print Services Facility/VM is a licensed program that can process a spool file or a file from the user's minidisk and send the print data to a printer that supports advanced function printing. PSF enables hardware and software enforced secure printing by providing:

- Header and trailer separator page labeling using overlays
- Random security numbers for print jobs (included on separator pages)
- Human-readable data page labeling (at the top and bottom of each page) using overlays

For additional information on PSF/VM Printer support, see *VM/ESA C2/B1 Trusted Facility Manual for VM/ESA with RACF* or *PSF/VM Security Guide*.

VM RSCS 3270 Secure Printing

VM RSCS 3270 Secure Printing provides 3270 print services on a local z/VM system, transferring print data between the local system and remote 3270 printers by providing:

- Controlling and auditing of secure printers within the trusted computing base.
- Enabling users to use these printers.
- Providing identification labels and random security numbers for each job printed by a secure printer.

For additional information on VM RSCS 3270 secure printing, see *VM/ESA C2/B1 Trusted Facility Manual for VM/ESA with RACF* and *RSCS 3270 Secure Printing: Installing, Managing, and Using*.

Operating at the C2 or B1 Level

RACF 1.9.2 for VM with VM/ESA Version 1 Release 2.0 and a set of related products is designed to meet U. S. Department of Defense (DoD) C2 and B1 levels of security criteria as documented in *VM/ESA C2/B1 Trusted Facility Manual for VM/ESA with RACF.* New functions added to RACF since RACF 1.9.2 are also designed to meet C2 and B1 levels of security, with the following exceptions:

- When secured signon authentication is used from a remote node to authenticate to the VM host, the host system does not meet B1 or C2 criteria.
- RACF support of OpenExtensions for z/VM is designed to meet C2 security criteria.
- RACF support of the shared file system (SFS) is designed to meet C2 security criteria.

For more information about operating at a C2 or B1 level on VM, see *VM/ESA C2/B1 Trusted Facility Manual for VM/ESA with RACF*.

For information for general users of an evaluated B1 or C2 system, see *VM/ESA C2/B1 Security Features User's Guide for VM/ESA with RACF*.

Appendix A. When Problems Occur

The information in this appendix is provided for your use in debugging problems with your profile definitions. It is designed to help you determine how your current RACF options and profiles work for you. For example, if users have access to resources that they should not have, or if users do not have access to resources that they should have, this section might be helpful in determining how to correct the problem.

Note: If you have a problem that you suspect is caused by RACF, not by your use of RACF, see *RACF Diagnosis Guide* for information on correcting the problem.

Checklist: Resolving Problems When Access Is Denied Unexpectedly

When a user or job requires access to a protected resource, and RACF denies the requested access, you will often have to analyze the problem before deciding what action to take. Here are some basic steps to take when analyzing access problems:

- First, get the complete text of the ICH408I message that RACF issued when denying the access. Also, take note of any other messages that were issued. The ICH408I message often indicates what profile RACF checked, and what class the profile was in, when denying access. Turn to *RACF Messages and Codes* for a description of the ICH408I message.

  Note: List the profile you believe should have given access. If it contains an *, %, or &, make sure it is also followed by a “G.” If you do not have the “G,” it isn't generic and would not be granting any access.

- If the ICH408I message indicates that access was denied because of a revoked user ID, you may want to resume that user ID.

- If the ICH408I message indicates that access was denied because of a profile, check the profile listing to make sure the user or job should have access. You should check not only the UACC and access lists, but the security classification of the resource profile and the user. Also, please note that installation exits (both RACF exits and certain exits in products that call RACF) can affect a user's access to resources. To check the user's access to the resource, ask the user who had the problem to list the profile protecting the resource.

  - For SFS files, the user can issue the LFILE command.
  - For SFS directories, the user can issue the LDIRECT command.
  - For general resources, the user can issue the RLIST command.

  In the listing, have the user check the YOUR ACCESS field.

- If the user cannot issue the LIST command, do it yourself. In the listing supplied by RACF, the following fields can, by themselves, deny access to a user:

  - The security level or security category, or both
  - The security label
  - The standard access list (ID and ACCESS)
  - UACC

- If the profile listing indicates that the user or job should have access, and the profile is in a class for which SETROPTS RACLIST processing or SETROPTS GENLIST processing is in effect, make sure that any in-storage profiles are refreshed by doing the following:

  - If the resource is protected by a generic profile in a class that is not RACLISTed or GENLISTed, ask the user to log off and log on again. This refreshes the user's copy of the profile.

  - Issue the SETROPTS GENERIC(*class-name*) REFRESH or SETROPTS RACLIST(*class-name*) REFRESH command again.

SETROPTS REFRESH Processing on Shared Systems

The refresh operation for SETROPTS processing applies only to the system (VM or MVS) on which you issue the SETROPTS command. If your installation has two or more systems sharing a RACF database, you must issue the SETROPTS command on all systems to have the refresh done on all systems. In a multiple RACF service machine environment, you must also issue the SETROPTS command to each service machine that shares the RACF database. (To coordinate the command across multiple RACF service machines, refer to the RAC command information in *RACF Command Language Reference*.)

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

- If access is being denied to a BFS file, use the **ls** command to display the file's permission bits and the file owner UID and GID. The user who has been denied access can issue the **id** command to display what the system is using as his or her UID, GID, and supplementary GID list. From this information, you can determine if RACF should allow access or not. If RACF is correctly denying access, you can change the file's permission bits using the **chmod** command to give the user access, if appropriate. If RACF should be allowing access based on the information shown, contact your IBM support center. The OpenExtensions shell commands **chmod**, **id**, and **ls** are documented in *OpenExtensions for z/VM: Command Reference.*
- You can use audit records to gather information that you would not otherwise have. In particular, you can request that audit records be generated for all accesses to protected resources, or for only failed accesses. You can also request that audit records be kept for a particular user. With the auditing in effect, you can use the RACF SMF data unload utility or the RACF report writer to view the access requests associated with the access requests.

  Note: In some cases (such as some resources in the OPERCMDS class), a RACROUTE request from a resource manager can include a “log string,” which is a string of characters to be placed in the SMF record if the access is audited. The log string can be useful in determining what kind of action the user was taking. For example, the log string might include the exact operator command, as the operator issued it.

Checklist: Resolving Problems When Access Is Allowed Unexpectedly

You may see occurrences when a user or job obtains access to a protected resource and you believe that the user should not have that access. There are many reasons why this could happen. The following checklist can eliminate some of them:

- Make sure that RACF is active, and that message ICH520I has been issued for this IPL. (To see if RACF is active, issue a RACF command, such as the LISTUSER command.)

  Note: Even though RACF is active, resource managers for which external security is optional (such as SFS) might not be using RACF. You must ensure that such resource managers are calling RACF. Also, there may be other options that control which general resource classes are protected by RACF. Therefore, you must make sure that the resource manager is calling RACF for the particular resource class.

- For VM minidisks, check to see if there is an entry in the global minidisk table for the minidisk in question. An entry in the global minidisk table can allow access when a profile protecting the resource would deny access. Global disk table entries are defined in HCPRWA using the GLBLDSK macro. See *RACF Macros and Interfaces* for details.
- If the resource is a general resource, make sure the general resource class is active. For example, for VM minidisks, make sure the VMMDISK class is active. To do this, issue the SETROPTS LIST command.
- Check for a global access checking table entry for the resource. An entry in the global access checking table can allow access when a profile protecting the resource would deny access. For example, for minidisks, issue the following:

  ```
  RLIST GLOBAL VMMDISK
  ```

  Note: Do not ignore the presence of entries containing &RACGPID or &RACUID.
- If the profile is a generic profile, use the SETROPTS LIST command to ensure that generic profile checking is in effect for the class.
- Make sure that control is on for the VM event that is involved. For example, for accesses to other user's minidisks, make sure that control for LINK is on. To do this, issue the following command:

  ```
  SETEVENT LIST
  ```

  Also, if a USERSEL profile exists for the user, check to make sure that control for the command is turned on in the USERSEL profile. To do this, issue the following command:

  ```
  SETEVENT LIST USERSEL.userid
  ```
- Make sure you know which profile is actually protecting the resource. For example, a more specific profile might actually be used instead of the generic profile you believe protects the resource. The more specific profile might grant the user the access. To do this, issue the LDIRECT, LFILE, or RLIST command, specifying the resource name.
- Check the user's access to the resource. You can do this in two ways:
  - Ask the user to list the profile protecting the resource. For example, the user can issue the LDIRECT, LFILE, or RLIST command, specifying the resource name. Have the user check the YOUR ACCESS field in the profile listing. If this field indicates that the user or job should have access, use the steps described in “Authorizing Access to Resources Protected by RACF Profiles” on page 307 to analyze the profile for reasons why the user or job has that access.
  - If the user cannot issue the LIST command, do it yourself. In the listing supplied by RACF, the following fields can, by themselves, allow access to a user:
    - For SFS files and directories, the second qualifier
    - The standard access list
    - The conditional access list
    - UACC

- WARNING

  If list-of-groups processing is in effect on your system, check to see if a group of which the user is a member is in the access list. Check both the standard access list and the conditional access list. Also, note that installation exits (both RACF exits and certain exits in products that call RACF) can affect a user's access to resources.

- If your analysis of the protecting profile shows that the user should not have access, continue with the following checks.
- If the SETROPTS RACLIST processing or SETROPTS GENLIST processing is in effect, make sure that any in-storage profiles are refreshed.
  - If the resource is protected by a generic profile in a class that is not RACLISTed or GENLISTed, ask the user to log off and log on again. This refreshes the user's copy of the profile.
  - Issue the SETROPTS GENERIC(*class-name*) REFRESH or SETROPTS RACLIST(*class-name*) REFRESH command again.

SETROPTS REFRESH Processing on Shared Systems

The refresh operation for SETROPTS processing applies only to the system (VM or MVS) on which you issue the SETROPTS command. If your installation has two or more systems sharing a RACF database, you must issue the SETROPTS command on all systems to have the refresh done on all systems. In a multiple RACF service machine environment, you must also issue the SETROPTS command to each service machine that shares the RACF database. (To coordinate the command across multiple RACF service machines, refer to the RAC command information in *RACF Command Language Reference*.)

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

- For resources defined in the VMCMD, VMMDISK, VMNODE, and VMRDR classes, the SYSSEC macro in HCPRWA can change the return code from the RACF service machine before giving control back to z/VM. Check your SYSSEC settings to determine if the access should be allowed. See *RACF Macros and Interfaces* for details on the SYSSEC macro.
- You can use audit records to gather information that you wouldn't otherwise have. In particular, you can request that audit records be generated for all accesses to protected resources, or for only successful accesses. You can also request that audit records be kept for a particular user. With the auditing in effect, you can use the RACF SMF data unload utility or the RACF report writer to view the access requests associated with the access requests.

  Note: In some cases (such as some resources in the OPERCMDS class), a RACROUTE request from a resource manager can include a “log string,” which is a string of characters to be placed in the SMF record if the access is audited. The log string can be useful in determining what kind of action the user was taking. For example, the log string might include the exact operator command, as the operator issued it.

When Changes to Minidisk Profiles Take Effect

A change to a minidisk profile might not take effect immediately, particularly for users who have already linked to the minidisk. For example, if you remove a user from the access list, and that user has already linked to the minidisk, merely changing the user's access does not cause the minidisk to be detached from the user.

A change to a minidisk profile affects a user's access to the profile immediately in the following cases:

- If the user is not logged on. You can check to see if a user is logged on with the CP QUERY command:

  ```
  QUERY userid
  ```

- If the user is logged on (or disconnected) and has not yet linked to the minidisk. You can check to see if a user is linked to a minidisk by linking to the minidisk yourself, and issuing the CP QUERY LINKS command:

  ```
  QUERY LINKS virtual-address
  ```

  Note: Linking to the minidisk yourself requires that you have access to the minidisk profile.

If the user is logged on (or disconnected) and has linked to the minidisk, and you change the user's access, two situations could occur:

- If the profile is a discrete profile, the user's access changes after detaching the minidisk.
- If the profile is a generic profile, the user's access changes after:
  - The user detaches the minidisk, if it is currently attached, and one of the following occurs when the user:
    - Issues SETROPTS GENERIC(VMMDISK) REFRESH (required if SETROPTS GENLIST(VMMDISK) is active)
    - Issues RL VMMDISK *userid.addr*
    - Enters a RACF command session
    - Logs off and then logs back on.
  - The copy of the generic profile that is kept in virtual storage is changed.

  The copy of the generic profile is changed when the user logs off and on again or when the SETROPTS GENERIC REFRESH command is issued.

Authorization Checking for Resources Protected by RACF Profiles

This section includes the following information:

- “When Authorization Checking Takes Place and Why” on page 307
- “Authorizing Access to Resources Protected by RACF Profiles” on page 307
- “Authorizing Access to RACF-Protected Terminals” on page 314
- “Authorization Checking for RACROUTE REQUEST=FASTAUTH Requests” on page 315
- “Authorizing Access to RACF-Protected Applications” on page 315

- “Security Label Authorization Checking” on page 315

When Authorization Checking Takes Place and Why

When a user requests access to a RACF-protected resource (such as a minidisk), the resource manager issues a RACROUTE REQUEST=AUTH request (or the RACHECK macro[5]). Based on the specifications on the RACROUTE REQUEST=AUTH request, or *RACF authorization request,* RACF determines whether the requesting user is authorized to access the resource.

- If the user is authorized to the resource, then RACF returns a “successful” return code to the resource manager. The resource manager then allows the request to complete.
- If the user is not authorized to the resource, then RACF returns an “unauthorized” return code to the resource manager. The resource manager then fails the request.

  RACF issues a message indicating that the resource is not protected.
- If the resource is not protected (for example, if no profile exists for it), then RACF returns the default return code for the class. For general resource classes, the default return code is the “not protected” return code, unless otherwise specified in the class descriptor table entry for the class.

  If the “not protected” return code is issued, the resource manager then either fails or allows the request. Most resource managers allow the request.

  RACF issues a message indicating that the resource is not protected.

Notes:

1. SMF log records and/or messages may be generated, depending on the options in effect and whether RACF granted or denied access to the resource.
2. For spool data (protected by the VMRDR class), if the user ID of the requesting user is the user ID that created the spool file, RACF grants the request.
3. For VM events that are not controlled—on a system-wide basis or for particular users—RACF is not called for authorization.

Authorizing Access to Resources Protected by RACF Profiles

To perform authorization checking for RACF-protected resources, RACF makes the following checks. RACF stops processing when the request is granted or denied.

Notes:

1. Access to a minidisk may be granted by the global minidisk table prior to any of the following steps occurring. For more information on the global minidisk table, see *RACF Macros and Interfaces*.
2. Access to an SFS file or directory may be granted by the SFSAUTOACCESS option prior to any of the following steps occurring. See *RACF System Programmer's Guide* for more information.

For an illustration of the following steps, see Figure 28 on page 312 and Figure 29 on page 313.

[5] If the RACHECK macro is issued instead of RACROUTE, authorization processing begins at step 5 on page 308.

1. If the entry for the class in the RACF router table is missing or has NONE specified, RACF returns the “not protected” return code. This also occurs if the caller specified the REQSTOR and SUBSYS operands on the RACROUTE macro, did not also specify DECOUPL=YES or give the REQSTOR and SUBSYS information in the RACF router table entry.

2. If RACF is not active, RACF returns the “not protected” return code.

3. For general resource classes, if the class of the resource is not active, RACF returns the “not protected” return code.

4. If the RACF class must be RACLISTed (as specified in the class descriptor table) but is not currently RACLISTed, RACF returns the “not protected” return code.

5. The RACROUTE REQUEST=AUTH preprocessing exit (ICHRCX01) can grant or deny the request.

6. If the user ID in the RTOKEN for the resource matches the user ID in the UTOKEN for the user making a request, RACF grants the request.

7. If profiles for the class have not been brought into storage by RACLIST processing, and if global access checking is active for the class, RACF searches the global access table (unless the CSA or PRIVATE operand was specified on the RACROUTE REQUEST=AUTH request). If RACF finds a matching entry that allows access to the resource, RACF grants the request.

8. RACF looks for a profile in storage or in the RACF database: If no profile is found that protects the resource, RACF returns the default return code of the class. (See the entry for the class in the CDT (class descriptor table), described in *RACF Macros and Interfaces*.) Specifically, no profile is found in the following cases:

   - Profiles for the class have been brought into common storage, but no profile in common storage matches the resource name.

   - If profiles for the class have *not* been brought into common storage, RACF looks for a profile in the user's storage. If no matching profile is found there, RACF looks in the RACF database.

   Note: If you expect generic profiles to be used by RACF authorization checking, list their profile names using the SEARCH command. If the profile names listed by the SEARCH command are *not* followed by (G), then RACF does not treat them as generic profiles. To recover, take the following steps:

   a. Issue SETROPTS NOGENERIC(*class-name*).

   b. Issue SETROPTS NOGENCMD(*class-name*).

   c. Delete the profiles. (If the profiles have complicated specifications, such as long access lists, you might wish to define “dummy” profiles modeled on them before deleting them. Specify the FROM operand on the RDEFINE command.

   d. Issue SETROPTS GENERIC(*class-name*).

   e. Define the profiles again.

9. If your installation has activated the SECLABEL class, RACF performs security label authorization checking. For a complete description, see “Security Label

Authorization Checking" on page 315. If security label authorization checking succeeds, RACF authorization checking continues with step 11 on page 309.

10. If the SECLABEL class is *not* active, the SECDATA class is active, and the requested resource has a security level or security category specified, RACF makes two checks in the sequence described below.

    a. RACF compares the security level (SECLEVEL) in the user profile with the security level in the resource profile. If the resource has a higher security level than the user, or if the user has no security level, RACF denies the request.

       For a terminal session, RACF uses the lower of the user's SECLEVEL and the terminal's SECLEVEL when authorizing access to a resource. For example, if the user has a SECLEVEL of 100 and the terminal has a SECLEVEL of 50, RACF uses the terminal's SECLEVEL during authorization checking. Thus, in this case, the user cannot access, through the terminal, any resource with a SECLEVEL greater than 50. (If the terminal is not defined to RACF or is defined without a SECLEVEL, RACF uses the user's SECLEVEL to determine the resources that can be accessed.)

       If the security level check passes, authorization checking continues with the following check.

    b. RACF compares the list of security categories in the user profile with the list of security categories in the resource profile. If the resource profile contains a security category that is not in the user's profile, RACF denies the request.

       Unlike the security level check, RACF uses *only* the user's security category list for a terminal session.

    If both checks succeed, RACF continues authorization checking with step 11.

11. If users attempt to access their own resources, RACF grants the request.

    For SFS file and directories, if the user ID of the requesting user is the second qualifier of the file or directory name, RACF grants the request.

12. RACF checks the user's access authority in the standard access list. If the user is in the list and if the specified access authority is sufficient to allow access, RACF grants the request. If the user is in the list and if the specified access authority is *less than* the requested access, RACF continues processing at step 17 on page 310 (conditional access list checking). This prevents access based on ID(*), UACC, or the OPERATIONS attribute.

    This could happen if, for example, user JOE requests UPDATE access, and the standard access list includes ID(JOE) ACCESS(READ).

13. RACF determines whether the user has access to the resource because the user is a member of a group and the group is on the standard access list. Which group is used depends on whether list-of-groups processing is in effect. (List-of-groups processing is in effect if the SETROPTS command has been issued with the GRPLIST operand.) RACF determines which group to use according to the following rules:

    - If list-of-groups processing is not in effect, RACF uses only the user's current connect group.

- If list-of-groups processing is in effect, RACF finds all the groups to which the user is connected that are also in the access list. Of these groups, RACF uses the group that has the highest access authority to the resource. (For example, assume that a user is a member of groups A, B, and C. If group A has NONE access authority, group B has READ access authority, and group C has UPDATE access authority, RACF will use group C to determine the user's access.)

  If the highest access authority is sufficient to allow the requested access, RACF grants the request. If the highest group that was found in the list does not have the requested authority, RACF continues processing at step 17 (conditional access list checking). This prevents access based on ID(*), UACC, or the OPERATIONS attribute.

14. If a user ID of * is found on the standard access list, the current user is defined to RACF, and the access authority granted to * is sufficient to allow the requested access, RACF grants the request.

15. If the universal access authority (UACC) for the resource provides sufficient access authority for the requesting user to access the resource, RACF grants the request.

16. If the requesting user has the OPERATIONS attribute (or group-OPERATIONS if the resource is within the scope of that group) and OPERATIONS access is allowed for the class, RACF grants the request.

17. RACF checks the user's access authority in the conditional access list specified with WHEN(TERMINAL). If the user is in the list, if the user meets the specified condition (such as logged on at the specified terminal), and if the specified access authority is sufficient to allow access, RACF grants the request.

18. RACF determines whether the user has access to the resource because the user is a member of a group that meets a condition specified on the conditional access list specified with WHEN(TERMINAL). Which group is used depends on whether list-of-groups processing is in effect. (List-of-groups processing is in effect if the SETROPTS command has been issued with the GRPLIST operand). RACF determines which group to use according to the following rules:

    - If list-of-groups processing is not in effect, RACF uses only the user's current connect group.

    - If list-of-groups processing is in effect, RACF finds all the groups to which the user is connected that are also in the access list. Of these groups, RACF uses the group that has the highest access authority to the resource. (For example, assume that a user is a member of groups A and B. If A has READ access authority and B has UPDATE access authority, RACF will use group B to determine the user's access.)

    If the group to be used according to the preceding rules has sufficient access authority to allow the requested access, RACF grants the request. If the group is in the list and if the specified access authority is NONE, RACF denies the request.

19. If a user ID of * is found on the conditional access list specified with WHEN(TERMINAL), the current user is defined to RACF and meets the specified condition (such as logged on at the specified terminal), and the

access authority granted to * is sufficient to allow the requested access, RACF grants the request.

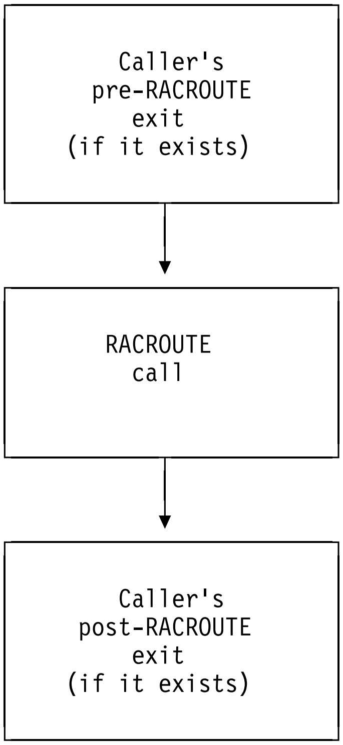
20. If the WARNING flag is ON in the profile (set using the WARNING operand on the ADDDIR, ADDFILE, ALTDIR, ALTFILE, RALTER or RDEFINE command), RACF grants the request.

21. The RACROUTE REQUEST=AUTH postprocessing exit (ICHRCX02) can grant or deny the request.

22. For resources defined in the VMCMD, VMMDISK, VMNODE, and VMRDR classes, the SYSSEC macro in HCPRWA can change the return code from the RACF service machine before giving control back to VM. See *RACF Macros and Interfaces* for details on the SYSSEC macro.

Authorization Checking

For a complete description of the numbered steps described in the following figures, see “Authorizing Access to Resources Protected by RACF Profiles” on page 307.


Caller's pre-RACROUTE exit (if it exists)
RACROUTE call
Caller's post-RACROUTE exit (if it exists)

Note: See the documentation for the calling product for information about exits that affect RACROUTE calls.

Figure 27. Process Flow of Callers of RACF for RACROUTE REQUEST=AUTH Requests

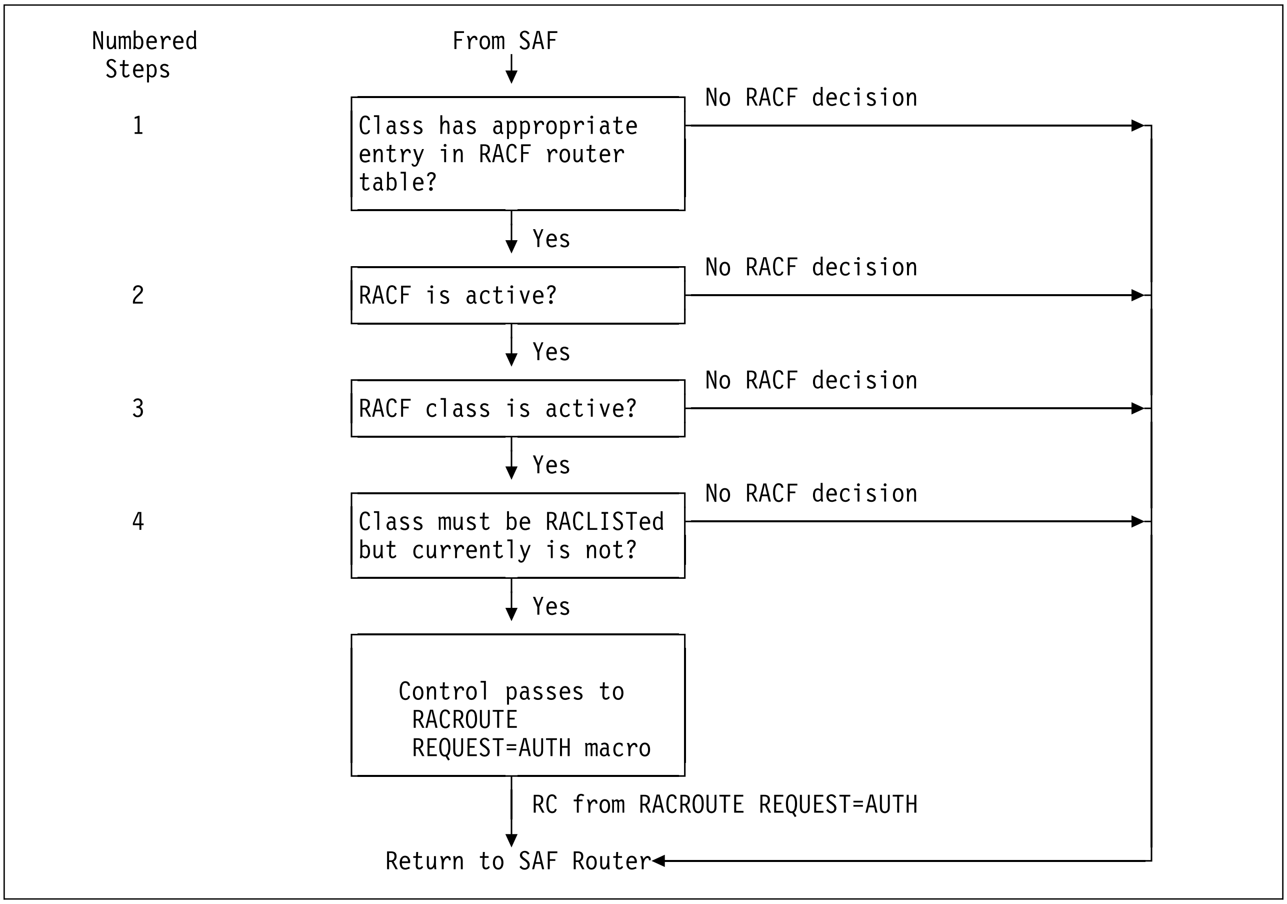

Numbered Steps
From SAF
1
Class has appropriate entry in RACF router table?
No RACF decision
Yes
2
RACF is active?
No RACF decision
Yes
3
RACF class is active?
No RACF decision
Yes
4
Class must be RACLISTed but currently is not?
No RACF decision
Yes
Control passes to RACROUTE REQUEST=AUTH macro
RC from RACROUTE REQUEST=AUTH
Return to SAF Router

Figure 28. Process Flow of RACF Router

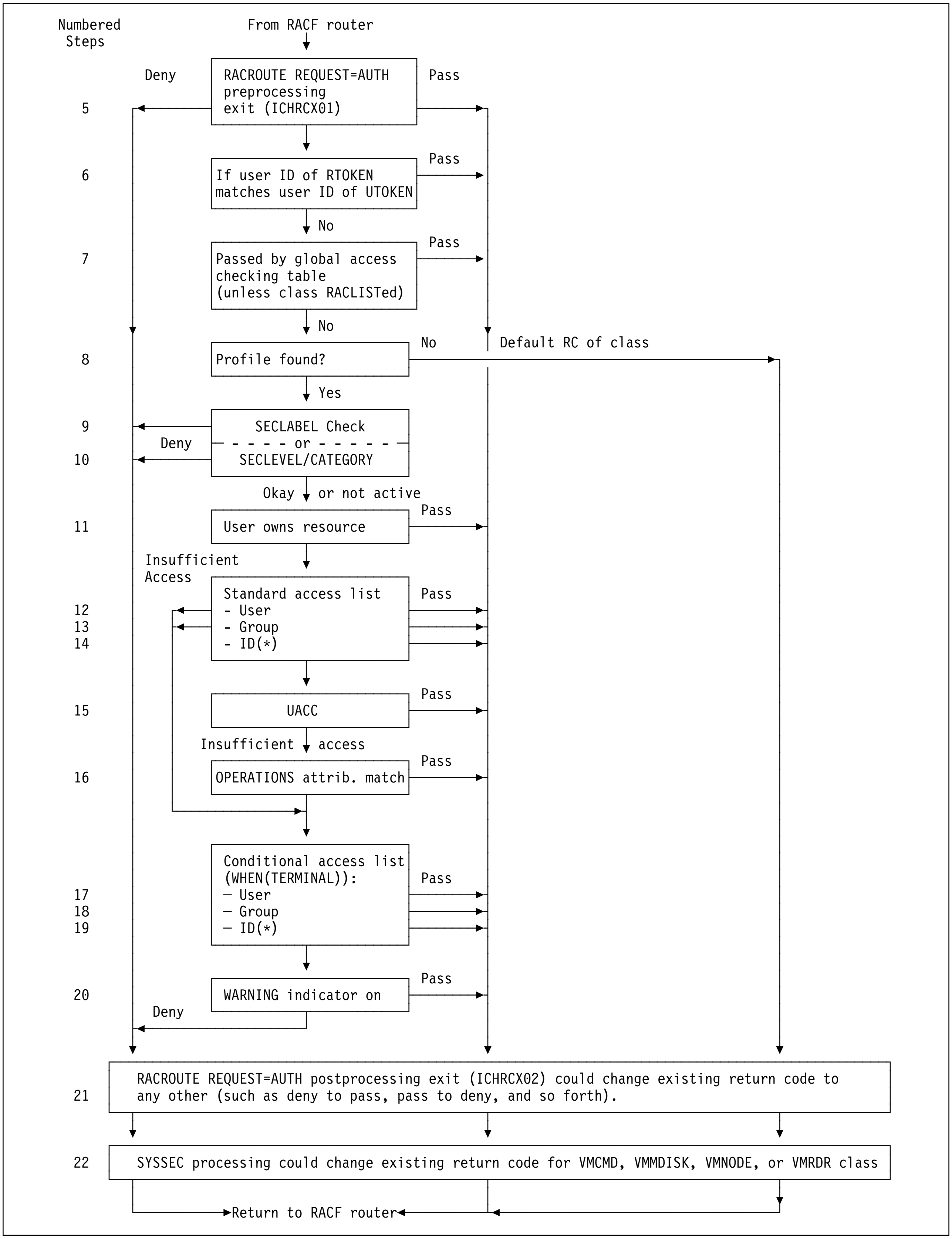

Numbered Steps
From RACF router
Deny
RACROUTE REQUEST=AUTH preprocessing exit (ICHRCX01)
Pass
5
6
If user ID of RTOKEN matches user ID of UTOKEN
Pass
No
7
Passed by global access checking table (unless class RACLISTed)
Pass
No
8
Profile found?
No
Default RC of class
Yes
9
SECLABEL Check
or
SECLEVEL/CATEGORY
Deny
10
Okay or not active
11
User owns resource
Pass
Insufficient Access
12
13
14
Standard access list
- User
- Group
- ID(*)
Pass
15
UACC
Pass
Insufficient access
16
OPERATIONS attrib. match
Pass
17
18
19
Conditional access list (WHEN(TERMINAL)):
— User
— Group
— ID(*)
Pass
20
WARNING indicator on
Pass
Deny
21
RACROUTE REQUEST=AUTH postprocessing exit (ICHRCX02) could change existing return code to any other (such as deny to pass, pass to deny, and so forth).
22
SYSSEC processing could change existing return code for VMCMD, VMMDISK, VMNODE, or VMRDR class
Return to RACF router

Figure 29. Process Flow of RACF Authorization Checking

Authorizing Access to RACF-Protected Terminals

When a RACF-defined user logs on to VM using a terminal protected by a profile in the TERMINAL or GTERMINL class and the TERMINAL class is active, RACF performs authorization checking to verify that the user is permitted use of the terminal. RACF performs this authorization checking during RACINIT processing at the same time as it performs user identification and verification.

RACF performs terminal authorization checking in the following sequence:

1. If your installation has activated the SECLABEL class, RACF performs security label authorization checking. For a complete description, see "Security Label Authorization Checking" on page 315. If security label authorization checking succeeds, RACF authorization checking continues with the next step.
2. If the requesting user has at least READ access authority to the terminal, RACF processing continues at step 5. If the user's access authority is NONE, RACF denies use of the terminal and stops terminal authorization checking.
3. If the requesting user's current connect group (or, if you activate list-of-groups checking, one of the user's other connect groups) has at least READ access authority to the terminal, RACF processing continues at step 5. If the group's access authority is NONE, RACF denies use of the terminal and stops terminal authorization checking.
4. If the profile has a universal access authority (UACC) of at least READ and your installation has not specified NOTERMUACC for the user's current connect group, RACF processing continues at step 5. Otherwise, RACF denies use of the terminal and stops terminal authorization checking.

   Note: For defined terminals, you can specify the universal access authority (UACC) with the RDEFINE or RALTER command. For undefined terminals, you can specify the universal access authority with the TERMUACC operand of the SETROPTS command.

   For VM systems, see "Restricting Specific Groups of Users to Specific Terminals" on page 183.
5. If your installation authorizes the use of the terminal on this particular day and time, RACF grants access to the terminal. (You can specify the terminal time and day-of-week restrictions with the RDEFINE and RALTER commands.) RACF also checks whether your installation has authorized the user to access the system on this particular day and time. (You can specify the user time and day-of-week restrictions with the ADDUSER and ALTUSER commands.)

Notes:

1. The RACROUTE REQUEST=AUTH and RACROUTE REQUEST=VERIFY preprocessing and postprocessing exit routines are available during terminal authorization checking.
2. Global access checking is not available during terminal authorization checking performed by RACROUTE REQUEST=VERIFY.
3. Profiles in the GTERMINL class are ignored unless SETROPTS RACLIST processing is in effect.

Authorization Checking for RACROUTE REQUEST=FASTAUTH Requests

Some resource managers have high performance requirements. In order to do resource authorization checking with RACF, they make use of RACF facilities to load all the profiles for a given class into storage. Once these profiles are in storage, instead of doing a normal authorization check, the resource managers can do a “fast” authorization check.

Fast authorization checking is different from normal authorization checking as follows:

- The privileged and trusted attributes are ignored.
- The global access checking table is not used.
- Security labels are not used.
- Security tokens are not used.
- No messages or SMF records are created. However, if the fast authorization processing determines that auditing is necessary, it indicates this to its caller. The caller can then issue a normal authorization request to cause SMF logging to be performed.

Authorizing Access to RACF-Protected Applications

You can control access to some applications by defining them to RACF as resources in the APPL class. When the user attempts to sign on the application, the application uses RACF to verify the user's identity and his authority to use that application. RACF does an authorization check to determine the user's authorization to the application.

- If there is a matching profile in the APPL class, RACF performs normal authorization checking as described in “Authorizing Access to Resources Protected by RACF Profiles” on page 307.
- If there is no matching profile in the APPL class, RACF allows the user to access the application.

Notes:

1. The RACROUTE REQUEST=AUTH and RACROUTE REQUEST=VERIFY preprocessing and postprocessing exit routines are available during application authorization checking.
2. Global access checking is not available during application authorization checking performed by RACROUTE REQUEST=VERIFY.

Security Label Authorization Checking

Note: This sequence of authorization checks begins in one of the sequences in “Authorization Checking for Resources Protected by RACF Profiles” on page 306.

When the SECLABEL class is active on your system, and a user or job requests access to a resource, RACF compares the security label of the user with the security label of the resource.

1. If the user requesting access does not have a security label and the resource does have a SECLABEL, RACF fails the request.

2. If the SETROPTS MLACTIVE(FAILURES) option is in effect and the resource does not have a security label associated with it, and the resource class requires security labels as defined in the class descriptor table, RACF fails the request.

   Note: If the SETROPTS MLACTIVE(FAILURES) option is in effect, users and jobs that do not have an associated security label cannot enter the system and no RACROUTE REQUEST=AUTH processing is ever done for them.

3. If the SETROPTS MLS(FAILURES) option is in effect, RACF makes one of the following tests, and fails the request if the test fails.

   - **Test for Read-Only Requests**

     If the request is only to read from the resource, the user's current security label must be greater than or equal to the security label of the resource. This is true when *both* the following are true:

     - The security level used to define the user's current security label is equal to or higher than the security level used to define the security label of the resource.
     - All the categories (if any) used to define the security label of the resource are in the user's current security label.

     When the user's current security label and the resource security label are related in this way, the user's security label is said to *dominate* the resource security label.

   - **Test for Read/Write Requests**

     If the request is to both read from and write to the resource, the user's current security label must have the same definition as the security label of the resource. (That is, the security level and categories that define the one security label must be the same as the security level and categories that define the other security label. The names of the security labels do not have to be the same.)

   - **Test for Write-Only Requests**

     If the request is to write only, the security label of the resource must be greater than or equal to the user's current security label. (Note that this is the opposite of read-only requests.)

     The MESSAGE, SMSG, MSGNOH, and WARNING commands are write-only requests, as are some VMCF and IUCV requests.

     Note: VM systems do not support write-only requests for tape volumes or minidisks. Therefore all such write requests are both read and write requests, and the security labels must be equal. Users can make write-only requests for SQL tables, but these resources are not protected by RACF.

4. If the SETROPTS MLS(WARNING) option is in effect, RACF makes the same checks as in step 3. If the access check fails on a read/write or write-only request because the user's current security label is greater than the the security label of the resource, RACF issues a warning message and grants the request.

5. If the SETROPTS NOMLS option is in effect, RACF makes the following tests and fails the request if the test fails:

- **Test for Read-Only and Read/Write Requests**

  The user's current security label must be greater than or equal to the security label of the resource. This is true when *both* the following are true:

  – The security level used to define the user's current security label is equal to or higher than the security level used to define the security label of the resource.

  – All the categories (if any) used to define the security label of the resource are in the user's current security label.

- **Test for Write-Only Requests**

  The user's current security label must be greater than or equal to the security label of the resource **or** the security label of the resource must be greater than or equal to the user's current security label. If this is true, then **either** of the following is true:

  – All the categories (if any) used to define the security label of the resource are in the user's current security label.

  – All the categories (if any) used to define the user's current security label are in the security label of the resource.

  When the user's current security label and the resource security label are not related in this way, they are said to be *disjoint*. Two security labels are disjoint when (1) the set of security categories that defines the first does not include the set of security categories that defines the second and (2) the set of security categories that defines the second does not include the set of security categories that defines the first.

Authorization Summary for SETROPTS MLS(FAILURES)

The following security label authorization rules apply when the SECLABEL class is active and SETROPTS MLS(FAILURES) is in effect:

- **Read Only:** The user's current security label must be greater than or equal to the security label of the resource.
- **Write Only:** The security label of the resource must be greater than or equal to the user's current security label.
- **Read/Write:** The user's current security label must have the same definition as the security label of the resource.

Note: A user is not allowed to write to a resource that has a security label that is less than the user's current security label. This inability to “write down” is enforced to ensure that a user does not declassify data.

Table 35 on page 318 describes the results of security label authorization when the SECLABEL class is active and SETROPTS MLS(FAILURES) is active.

Table 35. SECLABEL Authorization When SECLABEL Class and SETR MLS(FAILURES) Are Active

Relationship between User's Current SECLABEL and Resource SECLABEL	R/O request	R/W request	W/O request
User is greater than resource	Pass	Fail	Fail
Resource is greater than user	Fail	Fail	Pass
User equivalent to resource	Pass	Pass	Pass
Disjoint	Fail	Fail	Fail

Authorization Summary for SETROPTS NOMLS

The following security label authorization rules apply when the SECLABEL class is active and SETROPTS NOMLS is in effect:

- **Read Only:** The user's current security label must be greater than or equal to the security label of the resource.
- **Write Only:** The user's current security label.
  –or–
  The user's current security label must be greater than the security label of the resource.
  –or–
  The user's current security label must have the same definition as the security label of the resource.

  In other words, the user's current security label and the security label of the resource cannot be disjoint.
- **Read/Write:** The user's current security label must be greater than or equal to the security label of the resource.

Table 36 describes the results of security label authorization when the SECLABEL class is active and SETROPTS NOMLS or SETROPTS MLS(WARNING) is active.

Table 36. SECLABEL Authorization When SECLABEL Class and SETR NOMLS Are Active

Relationship between User's Current SECLABEL and Resource SECLABEL	R/O request	R/W request	W/O request
User is greater than resource	Pass	Pass (*)	Pass (*)
Resource is greater than user	Fail	Fail	Pass
User equivalent to resource	Pass	Pass	Pass
Disjoint	Fail	Fail	Fail

(*) For these items if SETROPTS MLS(WARNING) is active instead of NOMLS, a warning message (ICH408I) will be issued to the security console.

Authorization Summary for SETROPTS MLACTIVE

Table 37 describes the results of security label authorization when the SECLABEL class is active and either the user's or resource's security label is missing. The results vary depending on the SETROPTS MLACTIVE setting and whether or not the resource class being checked requires security labels. (The IBM-supplied class descriptor table defines which resource classes require security labels.)

Table 37. Effects of MLACTIVE Settings on Security Label Authorization

Environment	Missing User SECLABEL (Resource SECLABEL is present)	Missing Resource SECLABEL (User SECLABEL is present)	Missing User and Resource SECLABELs
MLACTIVE(FAILURES) and resource class requires SECLABELs	Fail (*)	Fail	Fail (*)
MLACTIVE(WARNING) and resource class requires SECLABELs	Fail	Pass and warning message sent to security console	Pass and warning message sent to security console
NOMLACTIVE and resource class requires SECLABELs	Fail	Pass	Pass
MLACTIVE(FAILURES) and resource class does not require SECLABELs	Fail (*)	Pass	Pass (*)
MLACTIVE(WARNING) and resource class does not require SECLABELs	Fail	Pass	Pass
NOMLACTIVE and resource class does not require SECLABELs	Fail	Pass	Pass

(*) For these items, the user has a missing SECLABEL and SETROPTS MLACTIVE(FAILURES) is in effect, so the user would not be allowed to log on to the system. If the user logged on before MLACTIVE(FAILURES) was activated, authorization requests will be passed/failed according to the entries in the table.

Special Access Rule for SPECIAL Users

If SETROPTS MLACTIVE(FAILURES) and SETROPTS MLS(FAILURES) are active, a RACF SPECIAL user who is logged on at the SYSHIGH SECLABEL is allowed to access resources that do not have a security label. RACF issues a warning message instead of failing the request. If these two SETROPTS options were turned on without proper preparation (assigning security labels to resources), this enables the security administrator to access resources on the system. To prevent violation of the no write-down rules, read-only access to resources is allowed with a warning message, but write access fails.

Relationships among SECLABEL, SETROPTS MLS(FAILURES), SETROPTS MLACTIVE(FAILURES) and SETROPTS MLQUIET

Table 38 shows the relationships of the security labels and the SETROPTS MLS, MLACTIVE(FAILURES) and MLQUIET options.

Table 38. Relationships among SECLABEL, SETROPTS MLS(FAILURES), SETROPTS MLACTIVE(FAILURES), and SETROPTS MLQUIET

SECLABEL	MLS (FAILURES)	MLACTIVE (FAILURES)	MLQUIET	Effect
Active	Off	Off	Off	RACF uses security labels and allows writing to a lower security label.
Inactive	Off	Off	Off	Security labels have no effect on authorization checking.
Active	On	Off	Off	RACF uses security labels and prevents writing to a lower security label (“no write down”).
Inactive	On	Off	Off	Security labels have no effect on authorization checking.
Active	Off	On	Off	Those resources required to have SECLABELS by the CDT, the DATASET class, and users must have security labels.
Inactive	On	Off	Off	Security labels have no effect on authorization checking.
Active	On	On	Off	All resources must be labeled, RACF uses security labels, and RACF prevents writing to a lower security label.
Inactive	On	On	Off	Security labels have no effect on authorization checking.
Either	Either	Either	On	All attempts to access the system or resources fail (unless the attempt is made by the trusted computing base, a security administrator, or a console operator).

Problems with User ID Authentication

This section includes the following information:

- “When Logon or Job Initialization Processing Takes Place and Why”
- “Logon/Job Initialization Processing” on page 321.

When Logon or Job Initialization Processing Takes Place and Why

When a user requests access to a VM system, the application controlling the user's access can issue the RACROUTE REQUEST=VERIFY or REQUEST=VERIFYX macro.

On VM, some of the places RACROUTE REQUEST=VERIFY requests occur are:

- When users issue the CP LOGON command
- When users submit batch jobs through the VMBATCH facility.
- When service machines are autologged

- When a user enters a RACF command session.

Based on the specifications on the RACROUTE REQUEST=VERIFY request, RACF determines whether the requesting user is authorized to enter the system.

- If the user is authorized to enter the system, then RACF returns a “successful” return code (return code 0) to the application. The application then allows the request to complete.
- If the user is not authorized to enter the system, then RACF returns an “unauthorized” return code (other than 0) to the application. In general, the application then fails the request.

Notes:

1. The RACROUTE REQUEST=VERIFY preprocessing and postprocessing exit routines are available during RACROUTE REQUEST=VERIFY processing.
2. RACF authorization checks can be requested by RACF or the application (for example, to determine if a user is authorized to use a particular terminal). RACROUTE REQUEST=AUTH pre- and post-processing exits are available during this authorization processing.
3. SMF log records and/or messages can be generated (failures are always recorded; successes can be recorded if the application requests it on the RACROUTE REQUEST=VERIFY request).

Logon/Job Initialization Processing

When users cannot log on (or jobs cannot be initiated) or started procedures fail, check the following:

- For all types of users and jobs, check for an authorization message (such as ICH408I) that indicates the cause of the failure, such as:
  - User profile not defined
  - User ID revoked
  - Invalid or no password
  - Invalid group ID
  - Invalid or no security label (depending on RACF options)

  If the application's message does not clearly indicate the source of the problem, find the ICH408I message. The ICH408I message, issued by RACF, may provide more information.

  Notes:

  1. This message is either on the user's terminal or, for jobs, in the job log, or, if unavailable, it is on the security console or the system log. Also, equivalent information is in audit records generated by RACF. Some information might be in audit records generated by the caller of RACF.
  2. If the 408I message indicates that access was denied because of a revoked user ID, you may want to resume that user ID. Check if the user ID is associated with the started procedure. If there was a user ID associated with the started procedure, this started procedure could not have begun successfully. After you resume the user ID, you must restart the started procedure or re-IPL.
- RACROUTE REQUEST=VERIFY processing might do some RACF authorization checks for the user. Also, the caller of RACF, or initial EXECs or

procedures that are invoked automatically might require RACF authorization checking.

See Table 39 to see which resource classes could be checked from various types of sessions.

- Check if an installation exit is causing the problem. Candidates include:
  - Exits in the caller of RACF
  - The RACINIT exits

Table 39. Resource Classes Checked for Logon/Job Initialization Requests

Type of Session	Classes That Might Be Checked
VM logons	SECLABEL, SURROGAT, TERMINAL, VMMDISK, or other classes, depending on the user's VM logon procedure

Appendix B. Command Summary and Authority Required to Issue RACF Commands

Summary of Commands and Their Functions

RACF commands allow you to list, modify, add, and delete profiles for users, groups, connect entries, and resources. Table 40 shows, in alphabetic order, each of the commands and its functions, and the system(s) on which you can invoke it. For a complete description of the authority required to issue a command and its operands, see *RACF Command Language Reference*.

Table 40 (Page 1 of 4). Functions of RACF Commands

RACF Command	Command Functions	System
ADDDIR [1]	• RACF-protect by a discrete or generic profile one or more SFS directories.	VM
ADDFILE [1]	• RACF-protect by a discrete or generic profile one or more SFS files.	VM
ADDGROUP	• Define one or more new groups as a subgroup of an existing group. • On MVS, specify a model data set profile for a group. • On MVS, define default DFP information for a group. • On VM, define the OpenExtensions VM information for a group.	MVS, VM
ADDSD [2]	• RACF-protect one or more existing data sets. • RACF-define one or more data sets brought from another system where they were RACF-protected. • RACF-define generic DATASET profiles. • Create a new data set model profile.	MVS
ADDUSER	• Define one or more new users and connect the users to their default connect group. • On MVS, specify a model data set profile for a user. • On MVS, define CICS operator information. • On MVS, define default DFP information for a user. • On MVS, define the preferred national language. • On MVS, define default operator information. • On VM, define the OpenExtensions VM information for a user. • On MVS, define default TSO logon information for a user. • On MVS, define default work attributes.	MVS, VM
ALTDIR [1]	• Change a discrete or generic SFS directory profile.	VM
ALTDSD [2]	• Change one or more discrete or generic DATASET profiles. • Protect a single volume of a multivolume, non-VSAM DASD data set. • Remove protection from a single volume of a multivolume, non-VSAM DASD data set.	MVS
ALTFILE [1]	• Change a discrete or generic SFS file profile.	VM
ALTGROUP	• Change the information in one or more group profiles (such as the superior group, owner, or model profile name). • On MVS, change or delete the default DFP information for a group. • On VM, add, change, or delete the information for an OpenExtensions VM group.	MVS, VM
ALTUSER	• Change the information in one or more user profiles (such as the owner, universal access authority, or security level). • Revoke or reestablish one or more users' privileges to access the system. • Specify logging of information about the user, such as the commands the user issues. • On MVS, change or delete CICS operator information. • On MVS, change or delete the default DFP information for a user. • On MVS, change the preferred national language. • On MVS, change or delete the default operator information. • On VM, add, change, or delete the information for an OpenExtensions VM user. • On MVS, change or delete the default TSO logon information for a user. • On MVS, change or delete the default work attributes.	MVS, VM
CONNECT	• Connect one or more users to a group. • Modify one or more users' connection to a group. • Revoke or reestablish one or more users' privileges to access the system.	MVS, VM
DELDIR [1]	• Remove RACF-protection from one or more SFS directories.	VM

Table 40 (Page 2 of 4). Functions of RACF Commands

RACF Command	Command Functions	System
DELFILE [1]	• Remove RACF-protection from one or more SFS files.	VM
DELDSD [2]	• Delete one or more discrete or generic DATASET profiles. • Delete a discrete DATASET profile for a tape data set, while retaining the data set name in the TVTOC. • Remove a data set profile, but leave the data set RACF-indicated, when moving a RACF-protected data set to another system that has RACF.	MVS
DELGROUP	• Delete one or more groups and their relationship to the superior group.	MVS, VM
DELUSER	• Delete one or more users and remove all their connections to RACF groups.	MVS, VM
DISPLAY [3]	• Display users signed on to a RACF subsystem.	MVS
END	• Terminate a RACF command session.	VM
HELP	• Display the function and proper syntax of RACF commands. • Display an explanation of command-related messages.	MVS, VM
LDIRECT [1]	• List the details of one or more discrete or generic DIRECTRY profiles, including the users and groups authorized to access an SFS directory.	VM
LFILE [1]	• List the details of one or more discrete or generic FILE profiles, including the users and groups authorized to access the SFS file.	VM
LISTDSD [2]	• List the details of one or more discrete or generic DATASET profiles, including the users and groups authorized to access the data sets. • Determine the most specific matching generic profile for a data set.	MVS
LISTGRP	• List the details of one or more group profiles, including the users connected to the group. • List only the information contained in a specific segment (RACF, DFP, or OVM) of the group profile.	MVS, VM
LISTUSER	• List the details of one or more user profiles, including all the groups to which each user is connected. • List only the information contained in a specific segment (for example, DFP or OVM information) of a user profile.	MVS, VM
PASSWORD	• Change one or more users' passwords. • Change one or more users' password change interval. • Reset one or more users' passwords to a known default value.	MVS, VM
PERMDIR [1]	• Give or remove authority to access an SFS directory to specific users or groups. • Change the level of access authority to a directory for specific users or groups. • Copy the list of authorized users from one directory profile to another. • Delete an existing access list.	VM
PERMFILE [1]	• Give or remove authority to access an SFS file to specific users or groups. • Change the level of access authority to a file for specific users or groups. • Copy the list of authorized users from one file profile to another. • Delete an existing access list.	
PERMIT	• Give or remove authority to access a resource to specific users or groups. • Change the level of access authority to a resource for specific users or groups. • Copy the list of authorized users from one resource profile to another. • Delete an existing access list.	MVS, VM
RACF	• Begin a RACF command session.	VM
RALTER	• Change the discrete and/or generic profiles for one or more resources whose class is defined in the class descriptor table. • Maintain the global access checking tables. • Maintain security category and security level tables.	MVS, VM
RDEFINE	• RACF-protect by a discrete and/or generic profile one or more resources whose class is defined in the class descriptor table. • Define the global access checking tables. • Define security category and security level tables.	MVS, VM
RDELETE	• Remove RACF-protection from one or more resources whose class is defined in the class descriptor table. • Delete the global access checking tables. • Delete the security category and security level tables.	MVS, VM

Table 40 (Page 3 of 4). Functions of RACF Commands

RACF Command	Command Functions	System
REMOVE	• Remove one or more users from a group and assign a new owner for any group data sets owned by the users.	MVS, VM
RLIST	• List the details of discrete and/or generic profiles for one or more resources whose class is defined in the class descriptor table.	MVS, VM
RVARY [4]	• Dynamically deactivate and reactivate the RACF function. • Dynamically deactivate and reactivate the RACF primary and backup database. • Switch the primary and backup RACF databases. • Deactivate resource protection, for any resource whose class is defined in the class descriptor table, while RACF is deactivated. • Select operational mode when RACF is enabled for sysplex communication.	MVS, VM
SEARCH	• List the RACF profile names that meet a search criteria for a class of resources. • Create a CLIST of the RACF profile names that meet a search criteria for a class of resources.	MVS, VM
SETEVENT	• Change the auditing or controlling of VM events. • Prevent users from issuing the DIAL, UNDIAL, and MESSAGE commands before logging on to the system. • Display the current status for VM events.	VM
SETRACF [5]	• Deactivate and reactivate RACF.	VM
SETROPTS	Dynamically set system-wide options relating to resource protection, specifically: For both MVS and VM systems: • Choose the resource classes that RACF is to protect. • Gather and display RACF statistics. • Set the universal access authority (UACC) for terminals. • Specify logging of certain RACF commands and events. • Permit list-of-groups access checking. • Display options currently in effect. • Enable or disable generic profile checking on either a class-by-class or system-wide level. • Control user password syntax rules. • Establish password syntax rules. • Activate password processing for checking previous passwords, limit invalid password attempts, and warn of password expiration. • Control global access checking for selected individual resources and/or generic names with selected generalized access rules. • Set the passwords for authorizing use of the RVARY command. • Initiate refreshing of in-storage generic profile lists and global access checking tables. • Enable or disable shared generic profiles for general resources in common storage. • Enable or disable shared profiles through RACLIST processing for general resources. • Activate or deactivate auditing of access attempts to RACF-protected resources based on installation-defined security levels. • Activate enhanced generic naming.	MVS, VM
	For MVS systems only: • Control the use of automatic data set protection (ADSP). • Activate profile modeling for GDG, group, and user data sets. • Activate protection for data sets with single-level names. • Control logging of real data set names. • Control the job entry subsystem (JES) options. • Activate tape data set protection. • Control whether or not data sets must be RACF-protected. • Control the erasure of scratched DASD data sets. • Activate program control.	MVS
SIGNOFF [3]	• Sign off users from a RACF subsystem.	MVS
SMF [6]	• On VM, restart SMF recording. • On VM, switch to the alternate SMF file.	VM

Table 40 (Page 4 of 4). Functions of RACF Commands

RACF Command	Command Functions	System
SRDIR [1]	• List the SFS directory profile names that meet search criteria.	VM
SRFILE [1]	• List the SFS file profile names that meet search criteria.	VM

Notes:

1. ADDDIR, ADDFILE, ALTDIR, ALTFILE, DELDIR, DELFILE, LDIRECT, LFILE, PERMDIR, PERMFILE, SRDIR, and SRFILE are RACF commands that operate in a VM environment. However, they can be used on MVS to maintain a RACF data base that is shared between VM and MVS.
2. ADDSD, ALTDSD, DELDSD, and LISTDSD are RACF commands that operate in an MVS environment. However, they can be used on VM to maintain a RACF data base that is shared between VM and MVS.
3. The DISPLAY and SIGNOFF commands can only be issued as operator commands. These commands perform RACF operations in a RACF subsystem.
4. The RVARY command can be issued as either a RACF operator command or a RACF TSO command. If this command is issued as an operator command, it operates in a RACF subsystem.
5. You can issue the SETRACF command only from a RACF service machine. A RACF service machine can run disconnected, thereby allowing a secondary console to issue this command.

   By default, RACF sets up the OPERATOR as the secondary console for a RACF service machine; the OPERATOR can issue the command to deactivate RACF. For example,

   ```
   SEND RACFVM SETRACF INACTIVE
   ```

   If you issue SETRACF for any RACF service machine in a multiple service machine environment, it will apply to all service machines.
6. SMF is only issued with SMSG.

Summary of Authorities and Commands

This section summarizes the attributes and authorities that can be assigned to users, and the RACF commands and operands that can be issued for each authority. There are eight tables which give information in four major categories: user attributes, group authorities, access authorities, and miscellaneous authorities.

Table 41. Commands and Operands You Can Issue If You Have the SPECIAL or group-SPECIAL Attribute

On VM and MVS:	
ADDGROUP	with all operands
ADDUSER	with all operands, but for group-SPECIAL user only when user also has CLAUTH(USER)
ALTGROUP	with all operands
ALTUSER	with all operands except UAUDIT or NOUAUDIT
CONNECT	with all operands
DELGROUP	with all operands
DELUSER	with all operands
LISTGRP	with all operands
LISTUSER	with all operands
PASSWORD	with all operands
PERMIT	with all operands
RALTER	with all operands except GLOBALAUDIT
RDEFINE	with all operands
RDELETE	with all operands
REMOVE	with all operands
RLIST	with all operands
SEARCH	with all operands
SETROPTS	with all operands except AUDIT, NOAUDIT, CMDVIOL or NOCMDVIOL, APPLAUDIT, NOAPPLAUDIT, LOGOPTIONS, OPERAUDIT, NOOPERAUDIT, SAUDIT, NOSAUDIT, SECLABELAUDIT, NOSECLABELAUDIT, and SECLEVELAUDIT, NOSECLEVELAUDIT, which require the AUDITOR attribute. User with group-SPECIAL attribute can issue only REFRESH GENERIC and LIST.
On VM:	
ADDDIR	with all operands
ADDFILE	with all operands
ALTDIR	with all operands except GLOBALAUDIT
ALTFILE	with all operands except GLOBALAUDIT
DELDIR	with all operands
DELFILE	with all operands
LDIRECT	with all operands
LFILE	with all operands
PERMDIR	with all operands
PERMFILE	with all operands
SETEVENT	with all operands. If you have both the SPECIAL and AUDITOR attributes and specify the REFRESH operand, you can refresh both auditing and controlling of VM events on your system. If you have only the SPECIAL attribute and specify the REFRESH operand, you can refresh only controlling of VM events on your system. If you specify a profile name on the REFRESH operand, you must also be the owner of the profile, or have ALTER authority to the profile. Users with the group-SPECIAL attribute cannot issue this command.
SMF	with all operands. Users with group-SPECIAL attribute cannot issue this command.
SRDIR	with all operands
SRFILE	with all operands
On MVS:	
ADDSD[1]	with all operands
ALTDSD[1]	with all operands except GLOBALAUDIT
DELDSD[1]	with all operands
LISTDSD[1]	with all operands

[1] This command applies to MVS systems. However, you can issue this command on a VM system to maintain a RACF database that is shared between MVS and VM.

Table 42. Commands and Operands You Can Issue If You Have the AUDITOR or group-AUDITOR Attribute

On VM and MVS:	
ALTUSER	only with UAUDIT or NOUAUDIT
LISTGRP	with all operands
LISTUSER	with all operands, lists UAUDIT or NOUAUDIT operand
RALTER	only with GLOBALAUDIT
RLIST	with all operands, lists GLOBALAUDIT option
SEARCH	with all operands
SETROPTS	only with AUDIT, NOAUDIT, CMDVIOL or NOCMDVIOL, LOGOPTIONS, OPERAUDIT, NOOPERAUDIT, SAUDIT, NOSAUDIT, SECLABELAUDIT, NOSECLABELAUDIT, SECLEVELAUDIT, NOSECLEVELAUDIT, LIST, or REFRESH GENERIC
On VM:	
ALTDIR	only with GLOBALAUDIT
ALTFILE	only with GLOBALAUDIT
LDIRECT	with all operands, lists GLOBALAUDIT option
LFILE	with all operands, lists GLOBALAUDIT option
SETEVENT	with only the LIST and REFRESH operands. If you have both the SPECIAL and AUDITOR attributes and specify the REFRESH operand, you can refresh both auditing and controlling of VM events on your system. If you have only the AUDITOR attribute and specify the REFRESH operand, you can refresh only auditing of VM events on your system. If you specify a profile name on the REFRESH operand, you must also be the owner of the profile, or have ALTER authority to the profile. Users with the group-AUDITOR attribute cannot issue this command.
SMF	with all operands. Users with group-AUDITOR attribute cannot issue this command.
SRDIR	with all operands
SRFILE	with all operands
On MVS:	
ALTDSD[1]	only with GLOBALAUDIT
LISTDSD[1]	with all operands, lists GLOBALAUDIT option

[1] This command applies to MVS systems. However, you can issue this command on a VM system to maintain a RACF database that is shared between MVS and VM.

Table 43. Commands and Operands You Can Issue If You Have the OPERATIONS or group-OPERATIONS Attribute

On VM and MVS:	
RLIST	with all operands except GLOBALAUDIT
SEARCH	with all operands
SETROPTS	only with REFRESH
On VM:	
LDIRECT	with all operands except GLOBALAUDIT
LFILE	with all operands except GLOBALAUDIT
SRDIR	with all operands
SRFILE	with all operands
On MVS:	
ADDSD[1]	when adding new profiles for group data sets
LISTDSD[1]	with all operands except GLOBALAUDIT

[1] This command applies to MVS systems. However, you can issue this command on a VM system to maintain a RACF database that is shared between MVS and VM.

Table 44. Commands and Operands You Can Issue If You Have the CLAUTH or group-CLAUTH Attribute

CLAUTH	
On VM and MVS:	
ADDUSER[1]	with all operands except OPERATIONS, NOOPERATIONS, SPECIAL, NOSPECIAL, AUDITOR or NOAUDITOR
ALTUSER[2]	only with CLAUTH or NOCLAUTH
RALTER[3]	only with ADDVOL
RDEFINE[4]	with all operands
SETROPTS[4]	only with REFRESH GLOBAL or REFRESH GENERIC

[1] This command applies when you have the CLAUTH attribute of USER and you either are the owner of a group, have JOIN authority in the default group specified in the command, or the profile is within the scope of a group in which you have the group-SPECIAL attribute.

[2] This command applies when you have the CLAUTH attribute for the class to be added or deleted, you are the owner of the user's profile, or the profile is within the scope of a group in which you have the group-SPECIAL attribute.

[3] This command applies when you have the CLAUTH attribute of TAPEVOL and you also have sufficient authority to issue the command.

[4] These commands apply when you have the CLAUTH attribute for the specified class.

Table 45. Commands and Operands You Can Issue If You Have a Group Authority

Group Authorities	**Commands and Operands You Can Issue If You Have This Authority**	
USE	***On VM and MVS:***	
	For group resources, the authority allowed the group.	
CREATE	***On MVS:***	
	ADDSD[1]	with all operands except NOSET
CONNECT	***On VM and MVS:***	
	ALTUSER	only with AUTHORITY, GROUP, or UACC
	CONNECT	with all operands except SPECIAL, NOSPECIAL, OPERATIONS, NOOPERATIONS, AUDITOR or NOAUDITOR
	LISTGRP	only with group name
	REMOVE	with all operands
	On MVS:	
	ADDSD[1,5]	with all operands except NOSET
JOIN	***On VM and MVS:***	
	ADDGROUP[2]	with all operands
	ADDUSER[3]	with all operands except OPERATIONS, SPECIAL or AUDITOR
	ALTGROUP[4]	only with SUPGROUP
	ALTUSER	only with AUTHORITY, GROUP, or UACC
	CONNECT	with all operands except SPECIAL, NOSPECIAL, OPERATIONS or NOOPERATIONS
	DELGROUP[2]	with all operands
	LISTGRP	only with group name
	REMOVE	with all operands
	On MVS:	
	ADDSD[1,5]	with all operands except NOSET

[1] This command applies to group data sets only.

[2] This command applies to the superior group.

[3] This command applies only if you have the JOIN group authority in the default group specified in the ADDUSER command and if you also have the CLAUTH(USER) attribute.

[4] This command applies to current and new superior groups. You may have JOIN authority in one group and be owner of or be connected with the group-SPECIAL attribute to another group.

[5] This command applies to MVS systems. However, you can issue this command on a VM system to maintain a RACF database that is shared between MVS and VM.

Table 46. Commands and Operands You Can Issue If You Have an Access Authority

Access Authorities	Commands and Operands You Can Issue If You Have This Authority	
NONE EXECUTE	none	
READ UPDATE CONTROL	***On VM and MVS:***	
	RLIST	with all operands except ALL or AUTHUSER
	SEARCH	with all operands
	On VM:	
	LDIRECT	with all operands except ALL or AUTHUSER
	LFILE	with all operands except ALL or AUTHUSER
	SRDIR	with all operands
	SRFILE	with all operands
	On MVS:	
	LISTDSD[1]	with all operands except ALL or AUTHUSER
ALTER	***On VM and MVS:***	
	PERMIT[2]	with all operands
	RALTER[2]	with all operands except ADDVOL[3], GLOBALAUDIT, or OWNER
	RDELETE[2]	with all operands
	RLIST[2]	with all operands
	On VM:	
	ALTDIR[2]	with all operands except GLOBALAUDIT or OWNER
	ALTFILE[2]	with all operands except GLOBALAUDIT or OWNER
	DELDIR[2]	with all operands
	DELFILE[2]	with all operands
	LDIRECT[2]	with all operands
	LFILE[2]	with all operands
	PERMDIR[2]	with all operands
	PERMFILE[2]	with all operands
	On MVS:	
	ALTDSD[1,2]	with all operands except GLOBALAUDIT, NOSET, or OWNER
	DELDSD[1,2]	with all operands except NOSET
	LISTDSD[1,2]	with all operands

[1] This command applies to MVS systems. However, you can issue this command on a VM system to maintain a RACF database that is shared between MVS and VM.

[2] This command applies to discrete profiles only.

[3] This command applies to ADDVOL operand only if you also have CLAUTH attribute for TAPEVOL.

Table 47. Commands and Operands You Can Issue If You Own a Profile

Owner of RACF Profile	**Commands and Operands You Can Issue If You Are the Owner**
Owner of user profile	***On VM and MVS:*** ALTUSER[1] only with user ID, NAME, OWNER, DFLTGRP, DATA, GRPACC, NOGRPACC, ADSP, NOADSP, REVOKE, RESUME, PASSWORD, NOPASSWORD, OIDCARD, NOOIDCARD, CLAUTH or NOCLAUTH DELUSER with all operands LISTUSER with all operands PASSWORD only with USER
Owner of group profile	***On VM and MVS:*** ADDGROUP[2] with all operands ADDUSER[3] with all operands except OPERATIONS, SPECIAL or AUDITOR ALTGROUP[4] with all operands ALTUSER only with GROUP, AUTHORITY or UACC CONNECT with all operands except SPECIAL, NOSPECIAL, OPERATIONS or NOOPERATIONS DELGROUP[5] with all operands LISTGRP with all operands REMOVE with all operands
Owner of resource profile	***On VM and MVS:*** PERMIT with all operands RALTER[6] with all operands except GLOBALAUDIT RDELETE with all operands RLIST with all operands SEARCH with all operands ***On VM:*** ALTDIR with all operands except GLOBALAUDIT ALTFILE with all operands except GLOBALAUDIT DELDIR with all operands DELFILE with all operands LDIRECT with all operands LFILE with all operands PERMDIR with all operands PERMFILE with all operands SRDIR with all operands SRFILE with all operands ***On MVS:*** ALTDSD[7] with all operands except NOSET or GLOBALAUDIT DELDSD[7] with all operands except NOSET LISTDSD[7] with all operands

[1] This command applies to CLAUTH or NOCLAUTH only if you have the CLAUTH attribute for the class to be added or deleted.

[2] This command applies to the superior group.

[3] This command applies to the default group specified and only if you have the CLAUTH attribute of USER.

[4] This command applies to current and new superior groups. You may have JOIN authority in one group and be owner of another group.

[5] This command applies to the superior group or group to be deleted.

[6] This command applies to the ADDVOL operand only when you also have CLAUTH attribute of TAPEVOL.

[7] This command applies to MVS systems. However, you can issue this command on a VM system to maintain a RACF database that is shared between MVS and VM.

Table 48. Commands and Operands You Can Issue for Miscellaneous Reasons

User ID Relationship	**Commands and Operands You Can Issue for Miscellaneous Reasons**
User ID is the current user	***On both VM and MVS:*** ALTUSER with DFLTGRP or NAME only LISTUSER with user ID only PASSWORD with INTERVAL or PASSWORD only
User ID is the high-level qualifier of a data set name (or a qualifier supplied by a command installation exit)	***On MVS only:*** ADDSD with all operands ALTDSD with all operands except GLOBALAUDIT or OWNER DELDSD with all operands LISTDSD with all operands PERMIT with all operands SEARCH with all operands
None	***On VM and MVS:*** RVARY[1] with all operands
User ID is the second qualifier of an SFS directory name	***On VM:*** ADDDIR with all operands ALTDIR with all operands except GLOBALAUDIT or OWNER DELDIR with all operands LDIRECT with all operands PERMDIR with all operands SRDIR with all operands
User ID is the second qualifier of an SFS file name	***On VM:*** ADDFILE with all operands ALTFILE with all operands except GLOBALAUDIT or OWNER DELFILE with all operands LFILE with all operands PERMFILE with all operands SRFILE with all operands

[1] Although no special authority is needed to issue this command, the system operator must supply the appropriate RVARY password, as established by the SETROPTS command with the RVARYPW operand, to approve any change in RACF status.

Notices

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

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

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
USA

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

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

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

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

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

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
Mail Station P300
2455 South Road
Poughkeepsie, NY 12601-5400
USA

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

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

COPYRIGHT LICENSE:

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

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Trademarks

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

BookManager
CICS
CICS/ESA
DB2
DFSMS/MVS
DFSORT
DirMaint
eServer
IBM
IBMLink
IMS
MVS
MVS/DFP
MVS/ESA
OfficeVision
OpenExtensions
OS/390
Print Services Facility
PROFS
QMF
RACF
Redbooks
S/390
SP
SQL/DS
SystemView
VM/ESA
VTAM
z/OS
z/VM
zSeries

Tivoli, TME and NetView are trademarks of International Business Machines or Tivoli Systems, Inc. in the United States, other countries or both.

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

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

Index

Communicating Your Comments to IBM

Resource Access Control Facility
Security Administrator's Guide
Version 1 Release 10

Publication No. SC28-1340-15

If you especially like or dislike anything about this book, please use one of the methods listed below to send your comments to IBM. Whichever method you choose, make sure you send your name, address, and telephone number if you would like a reply.

Feel free to comment on specific errors or omissions, accuracy, organization, subject matter, or completeness of this book. However, the comments you send should pertain to only the information in this manual and the way in which the information is presented. To request additional publications, or to ask questions or make comments about the functions of IBM products or systems, you should talk to your IBM representative or to your IBM authorized remarketer.

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

If you are mailing a reader's comment form (RCF) from a country other than the United States, you can give the RCF to the local IBM branch office or IBM representative for postage-paid mailing.

- If you prefer to send comments by mail, use the RCF at the back of this book.
- If you prefer to send comments by FAX, use this number:
  - FAX: (International Access Code)+1+845+432-9405
- If you prefer to send comments electronically, use the following e-mail address:
  - mhvrcfs@us.ibm.com

Make sure to include the following in your note:

- Title and publication number of this book
- Page number or topic to which your comment applies

Optionally, if you include your telephone number, we will be able to respond to your comments by phone.

Reader's Comments — We'd Like to Hear from You

Resource Access Control Facility
Security Administrator's Guide
Version 1 Release 10

Publication No. SC28-1340-15

You may use this form to communicate your comments about this publication, its organization, or subject matter, with the understanding that IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you. Your comments will be sent to the author's department for whatever review and action, if any, are deemed appropriate.

Note: Copies of IBM publications are not stocked at the location to which this form is addressed. Please direct any requests for copies of publications, or for assistance in using your IBM system, to your IBM representative or to the IBM branch office serving your locality.

Today's date: ______________________________

What is your occupation?

Newsletter number of latest Technical Newsletter (if any) concerning this publication:

How did you use this publication?

[ ] As an introduction
[ ] As a reference manual
[ ] For another purpose (explain)

[ ] As a text (student)
[ ] As a text (instructor)

Is there anything you especially like or dislike about the organization, presentation, or writing in this manual? Helpful comments include general usefulness of the book; possible additions, deletions, and clarifications; specific errors and omissions.

Page Number: Comment:

Name

Address

Company or Organization

Phone No.

Reader's Comments — We'd Like to Hear from You
SC28-1340-15

Cut or Fold
Along Line

Fold and Tape | **Please do not staple** | Fold and Tape

NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

IBM Corporation
Department 55JA, Mail Station P384
2455 South Road
Poughkeepsie, NY 12601-5400

Fold and Tape | **Please do not staple** | Fold and Tape

Cut or Fold
Along Line

Program Number: 5741-A05

Printed in the United States of America
on recycled paper containing 10%
recovered post-consumer fiber.

SC28-1340-15