SC28-1340-1
File No. S370-34

Resource Access Control Facility (RACF) Security Administrator's Guide

Program Product

IBM

Second Edition (May, 1984)

This is a major revision of, and obsoletes, SC28-1340-0. See the Summary of Amendments following the Contents for a summary of the changes made to this manual. Technical changes or additions to the text and illustrations are indicated by a vertical line to the left of the change.

This edition applies to Version 1 Release 6 with the data security monitor of the program product RACF (Resource Access Control Facility), Program Number 5740-XXH, and to all subsequent releases and modifications until otherwise indicated in new editions or Technical Newsletters. Changes are made periodically to the information herein; before using this publication in connection with the operation of IBM systems, consult the latest *IBM System/370 Bibliography*, GC20-0001, for the editions that are applicable and current.

References in this publication to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM program product in this publication is not intended to state or imply that only IBM's program product may be used. Any functionally equivalent program may be used instead.

Publications are not stocked at the address given below. Requests for IBM publications should be made to your IBM representative or to the IBM branch office serving your locality.

A form for readers' comments is provided at the back of this publication. If the form has been removed, comments may be addressed to IBM Corporation, Information Development, Department D58, Building 920-2, PO Box 390, Poughkeepsie, N.Y., 12602. IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

Preface

This publication contains information for Version 1 Release 6, with the data security monitor, of the program product RACF (Resource Access Control Facility - Program Number 5740-XXH). This book is intended for security administrators, group administrators, and other administrators responsible for system data security and integrity. The readers of this book should be familiar with MVS.

RACF is a program product that provides access control by:

- Identifying and verifying system users
- Authorizing access to system resources
- Logging unauthorized attempts to enter the system and accesses to protected resources

The major topics in this publication are:

- Chapter 1, which provides an overview of RACF and various administrative information for the security and group administrator
- Chapter 2, which includes information on how to organize for RACF implementation
- Chapter 3, which includes information on how to define groups and users
- Chapter 4, which includes information on how to define resources
- Chapter 5, which includes information on how to select RACF options
- Chapter 6, which discusses operating considerations
- Chapter 7, which discusses RACF usage with IMS/VS
- Chapter 8, which discusses RACF usage with CICS/VS

There is one appendix:

- Appendix A includes a RACF command summary and examples of command usage

Corequisite Publications:

- *RACF General Information Manual*, GC28-0722
- *RACF Command Language Reference*, SC28-0733

Related Reading:

These publications are referenced in the text and/or contain information related to using RACF. If a shortened title is used in this publication, the short title appears in parentheses following the publication number.

System Programming Library: Resource Access Control Facility (RACF), SC28-1343 (*SPL: RACF*)

Resource Access Control Facility (RACF) Auditor's Guide, SC28-1342 (*Auditor's Guide*)

MVS Security, GC28-1400

OS/VS2 JCL, GC28-0692 (*JCL*)

MVS/Extended Architecture JCL, GC28-1148 (*JCL*)

OS/VS2 System Programming Library: System Management Facilities (SMF), GC28-1030 (*System Management Facilities (SMF)*)

MVS/Extended Architecture System Programming Library: System Management Facilities (SMF), GC28-1153 (*System Management Facilities (SMF)*)

OS/VS2 TSO Command Language Reference, GC28-0646 (*TSO Command Language Reference*)

MVS/Extended Architecture TSO Command Language Reference, GD23-0259 (*TSO Command Language Reference*)

OS/VS Tape Labels, GC26-3795

MVS/Extended Architecture Tape Labels, GC26-4003

OS/VS2 Access Method Services, GC26-3841 (*Access Method Services*)

MVS/Extended Architecture Access Method Services Reference for VSAM CATALOGS, GC26-4075 (*Access Method Services*)

OS/VS2 MVS CVOL Processor, GC26-3864

MVS/Extended Architecture Catalog Users Guide, GC26-4075

Contents

Figures

Contents Directory

This page left blank

Summary of Amendments

Summary of Amendments
for SC28-1340-1
RACF Version 1, Release 6 with DSMON

This newsletter contains updates in support of the data security monitor (DSMON), as well as minor technical and editorial changes.

The data security monitor (DSMON) is a batch program that allows users with the AUDITOR attribute to obtain a set of reports that provide information about the current status of your installation's data security environment. As a RACF administrator, you might find these reports to be a useful tool both during the initial implementation of RACF and at regular intervals during normal operation.

Chapter 1. Introduction

The Need for RACF Protection

Over the past few years it has become much easier to create and access computerized data. No longer is access limited to a handful of highly skilled programmers; computerized data 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 making use of computer systems has increased dramatically. And now, more and more people are becoming increasingly dependent on their computer systems and on the data they store in these systems.

As the general computer literacy and the number of people making use of 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 security administrator, it is your job to ensure that your installation's data is properly protected. RACF is designed to help you do this, while imposing a minimum effort on you or your installation's end users.

How RACF Meets Security Needs

The RACF program product is designed to 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

- Is transparent, or at least relatively painless, to the majority of end users, and has little or no impact on an installation's current operation
- Is easy to install and maintain.

User Identification and Verification

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 *userid* and a system-encrypted *password* to perform its user identification and verification. When you or your delegate define a user to RACF, you assign a userid and temporary password. The userid identifies the person to the system as a RACF user. 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 userid-password combination.

Note: During terminal processing, operator identification card (OIDCARD) in place of or in addition to the password. (The OIDCARD information is also encrypted.) By requiring that a user not only knows the correct password but also furnishes the correct OIDCARD, you have increased assurance that the proper user has entered the userid.

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. 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 or your delegate are responsible for protecting the system resources (tape volumes, DASD data sets, terminals, and so on) and for issuing the authorities by which those resources are made available to users. RACF records your assignments in "profiles" stored in the RACF data set. RACF then refers to the information in the profiles to decide if a user should be permitted to access a system resource.

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 smooth functioning security system.

Because it can identify and verify the user 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 can identify the authorized accessors to the resource owners. In addition, you and/or your auditor can use these functions to detect possible security exposures or threats. The functions are:

- **Logging:** RACF writes records to SMF (system management facilities) 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 data set

  To list SMF records, you can use the RACF report writer. With this report writer, you can select RACF SMF records to be used to produce the reports.

- **Sending Messages:** RACF sends messages to the security console for the more serious violations, such as detected, unauthorized attempts to enter the system and, optionally, detected, unauthorized attempts to access RACF-protected resources or modify profiles on the RACF data set.

  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, RACF sends a warning message to the accessor requesting the person to contact you or the resource owner.

- **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 the installation with the ability to list the contents of the profiles in the RACF data set.

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.

In the case of TSO, where an individual user accesses the system through a terminal, the concept of individual user identity is fairly obvious. 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 alphameric *userid* that RACF associates with the user. Note, however, that a RACF user need not be an individual. For example, a userid can be associated with a started task. 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 userid 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. It is an administrative convenience.

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 associated with (connected to) any number of groups. Membership in, and authority in, these groups can be used to control the scope of a user's activity.

How Users and Groups Are Authorized to Access Resources

Basically, a user's level of authorization while operating in a RACF-protected system at any time is determined by a combination of three gating factors:

- User's **attributes**
- User's **group authorities**
- **Resource access authorities** associated with the various DASD data set and general 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: You can assign the SPECIAL, AUDITOR, OPERATIONS, CLAUTH, GRPACC, ADSP, and REVOKE attributes at the user level. These attributes are described in detail later in this chapter and in Chapter 3. When attributes are assigned at the user level, the privileges and restrictions apply across the entire system.
- Group-level Attributes: When an attribute is assigned at the group level, the scope of the privilege or restriction conveyed by the attribute is limited to the group to which it applies. The group-SPECIAL, group-AUDITOR, and

group-OPERATIONS attributes are discussed later in this chapter and in Chapter 3.

Group Authorities: Each user must be assigned to at least one group (called the default group). Within that group, or any other group to which the user may belong, the user may have responsibilities and privileges that differ from other members of the group. For example, the user may be responsible for adding new members to the group, while someone else may be responsible for maintaining resource protection profiles owned by the group. The security administrator or group administrator can assign a specific level of "group authority" to each user of a group. The USE, CREATE, CONNECT, and JOIN group authorities are described in detail later in this chapter and in Chapter 3.

Resource Access Authorities: The owner of a protected DASD data set or general resource (such as a disk volume, tape volume, IMS or CICS transaction or application) can grant or deny a user or group access to that resource by including the applicable resource access authority in the profile that controls access to that resource. The resource access authority determines to what extent the specified user or group can use the resource. The UACC universal access control authority, ALTER, CONTROL, UPDATE, READ, and NONE resource access authorities are described in detail later in this chapter and in Chapter 4.

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 restrict 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.)

There are five types of profiles:

- User profiles
- Group profiles
- User-to-group connect profiles, commonly called "connect" profiles
- Data set profiles
- General resource profiles

User, group, and connect profiles contain descriptions of the authorized users of a RACF-protected system; data set and 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 and the RACFRW (report writer) command, have Interactive System Productivity Facility (ISPF) entry panels and associated help panels. These panels make it easy to enter command options. (Output from the commands is in TSO line mode. If the TSO session manager is installed, you can use it to scroll through the output from the listing commands.) Note that use of the RACF ISPF panels requires that ISPF (Program Number 5668-960) and TSO/E Version 2 be installed at your location.

- **Generic Profiling:** RACF generic profiling allows you, your group administrators, and other users to define profiles that consolidate the security requirements of several similarly-named and similarly-accessed resources.

- **Protection of Installation-Defined Resources:** RACF protects DASD data sets and volumes, tape volumes, terminals, applications, and transactions defined to IMS/VS and CICS/VS. In addition, RACF permits the installation to protect installation-defined resources.

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

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 for this to happen. Unfortunately, human nature being as it is, 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. By making use of generic profiles and/or automatic data set protection (ADSP) and profile modeling, security and group administrators can 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 study.

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 lower, 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 systems 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.

- **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. The auditor, like you, can delegate authority.

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 for details), security and group administration tasks are performed, 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 default group; you can use the ADDSD command to protect a DASD data set, and so on. (See Appendix A for a summary of RACF commands and the attributes and authorities you need to use them.) The RACF commands include keywords, with which you specify the various user attributes, group authorities, and resource access authorities. RACF places the information gotten from the commands into various profiles (user, group, connect, data set, and general resource profiles), which it keeps in the RACF data set 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 keyword names; you need only fill in the appropriate information on the proper panels. See the *RACF Command Language Reference* for details on panel use.

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. Some of the commands you might use in your user-definition tasks follow. (Note that this list is not an exhaustive list of either RACF commands or command descriptions. For a more complete description, see Appendix A and/or the *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.

Some of the commands you might use in your group-definition tasks follow. (Note that this is not an exhaustive list of either commands or command descriptions. For a more complete description, see Appendix A and/or the *RACF Command Language Reference.*)

Commands for Group Administration

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

You can assign user attributes by specifying keywords 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 user level or at the group level. When assigned at the user level, attributes are effective globally for the entire RACF-protected system. When assigned at the group level, their effect is limited to the profiles of resources 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 1-1 shows an example of the scope of control of an attribute assigned at the group level.

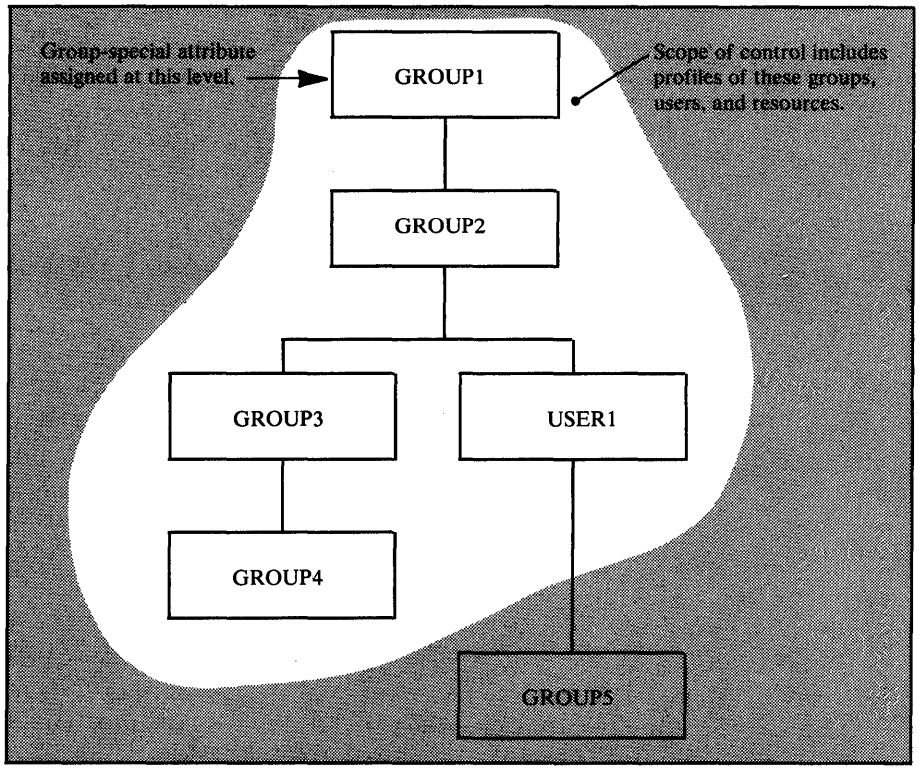


Figure 1-1. Scope of Control of an Attribute Assigned at the Group-Level

Figure 1-1 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. Figure 1-2 lists and describes attributes that can be assigned at the user and group level. For a more complete description, see Chapter 3.

User Attribute	Description
SPECIAL	The SPECIAL attribute gives the user full control over all the RACF profiles in the RACF data set when you assign it at the user level. At the user 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 user 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 the SMF data set for detecting changes and attempts to change the RACF data set 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 user 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-name authorization) attribute allows the user to define profiles for any of the classes specified by class-name. A class is a collection of RACF entities with similar characteristics. See Chapter 3 for a list of valid class-names.
GRPACC	Group data sets that are allocated by this user and protected by discrete profiles are automatically accessible to other users in the group. A group data set is a data set whose high-level qualifier (or RACF naming convention table-derived qualifier) is equivalent to 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.

Note: 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.

Figure 1-2. User Attributes

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 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 resource 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 members and new groups. See "Group Authorities" in Chapter 3 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 data set. A general description of user, group, and connect 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 (encrypted or masked)
- User attributes that are effective globally
- The name of the user's default group
- The name of a model profile to be used when new profiles are created
- Information about what logging is to be done for this user
- How often the user's password is to be changed

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
- The group authorities of each member
- The name of a model profile to be used when new groupname data set profiles are created.

The Connect Profile: A connect profile is created automatically whenever you define a new user to RACF and connect the user to a default group with the ADDUSER command, or whenever you connect a previously-defined RACF user to an existing RACF group with the CONNECT command. The connect profile contains:

- The name of the owner of the profile
- User attributes that are effective within the scope of the group
- Other information about the group

Protecting Resources

The resources RACF protects can be divided into two categories: DASD data sets (DATASET class) and general resources. Access to a protected resource is controlled by a *discrete* or *generic* profile that is created to represent that resource.

A discrete profile protects a single resource. You should create one for any resource that has unique access-authorization or logging and statistics-keeping requirements. (For example, for a single, sensitive data set residing on a particular volume.)

You can use a generic profile to protect a number of related resources -- that is, resources having similar access-authorization requirements and naming structures. Using a generic profile often requires a minimum amount of RACF profile maintenance on the part of a RACF administrator.

Discrete and generic profiles are discussed in more detail throughout the remainder of this chapter as they apply to data set and general resource definitions.

Protecting DASD Data Sets

RACF can protect cataloged and uncataloged non-VSAM data sets, VSAM data sets, data sets that have the same name but reside on different volumes, and generation data group (GDG) data sets.

RACF also protects data sets whether or not they are password protected. When both RACF protection and password protection are applied to a data set, access to the data set is determined only through RACF authorization checking. That is, password protection is bypassed. (RACF protection has an advantage over password protection: with RACF protection, only authorized users can access the data set; with password protection, any user who knows the password can access the data set. Also, users can run jobs more easily using RACF protection because the system operator is not prompted for data set passwords for RACF-protected data sets accessed during a job.)

To protect a DASD data set, a user invokes a TSO command that creates a data set profile and stores it in the RACF data set.

You can protect DASD data sets with *discrete profiles* or *generic profiles*. If a data set has unique access-authorization or logging requirements, you should define a discrete profile for it; if the requirements are for several data sets that share a common name structure, you can define a generic profile that applies to all of the data sets.

Discrete Profile: A discrete profile contains a description of the data set, including the authorized users, the access authority of each user, the location of the data set (device type and volume serial numbers), and other information, such as the number of accesses to the data set.

Generic Profile: A generic profile is similar to a discrete profile; however, it can be shared by many DASD data sets that have a similar naming structure, regardless of the units or volumes those data sets reside on. Also, data sets that are protected by generic profiles do not have to be defined individually to RACF.

If data management always call is not available on your system, the set of data sets protected by a generic profile is a subset of those protected when always call is available. That is, only data sets that are "RACF-indicated" (and do not have an associated discrete profile) can be protected by a generic profile on a system without always call. See "Protection Via Generic Profiles" in Chapter 4 for details.

Protecting General Resources

You can protect DASD volumes, tape volumes, IMS/VS transactions, IMS/VS transaction groups, applications, CICS/VS transactions, CICS/VS transaction groups, CICS/VS program specification blocks (PSBs), CICS/VS PSB groups, terminals, and any other resource classes that you define to RACF.

When you define a resource class, RACF places control information for the new resource class into a *class descriptor table.* The control information includes the resource class name, the syntax rules for the member names within the class, and the location of the auditing and statistics flags for the class. You must supply the necessary control information for any new classes you define. (RACF supplies the control information for the standard resource classes.)

When you define a new resource class, you may 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. (GIMS and TIMS are an example of a resource group class and its respective resource member class.)

RACF refers to the class descriptor table whenever a class-related decision (such as, "Should auditing be done for this class?") must be made. With the services provided by the class descriptor table, and with appropriate use of RACF authorization checking services (RACF macros), you can extend RACF protection to any part of the system.

When you use the RDEFINE command to define a resource other than a DASD data set to RACF, RACF builds a general resource profile. The general resource profile contains information about the resource, such as class name, resource name, and which attempts, successes or failures, are to be logged on the SMF data set. It also contains a list of all the users or groups that have been authorized (by a PERMIT command) to access the resource.

Selecting RACF Options

RACF options provide flexibility in the creation and administration of your RACF security system. RACF options control:

- Generic profile checking
- Statistics gathering
- Undefined terminal protection
- User password expiration
- Global access checking
- Single-level data set name processing
- ADSP
- Logging of RACF events
- Other RACF activities

When efficiently implemented, RACF options can effectively enhance performance and recovery.

You can select RACF options with the SETROPTS command, the data set name table, and the ICHSECOP RACF module. Chapter 5 describes the SETROPTS command options and the ICHSECOP options in more detail; Appendix A describes the naming convention table.

Tailoring RACF

You can tailor RACF to bypass security checking or to perform additional security 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 *SPL: RACF.* 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.)

RACINIT, RACHECK, and RACDEF Exits: The RACINIT, RACHECK, and RACDEF SVCs, respectively, perform the RACF user verification, access control, and dynamic resource definition functions. Preprocessing installation exits are available to tailor the parameters specified by the RACINIT, RACHECK, and RACDEF macro instructions or to perform any additional security checks. Postprocessing exits are available to override or modify results of RACF processing performed by the RACINIT, RACHECK, and RACDEF SVCs.

RACLIST Exits: The RACLIST SVC, used to build in-storage copies of general resource profiles, has two exits. The preprocessing/postprocessing exit is called before RACLIST processing to allow the installation to alter RACLIST processing options and after RACLIST processing to perform housekeeping.

FRACHECK Exits: The FRACHECK routine uses the resident profiles constructed by RACLIST to perform authorization checking. FRACHECK has a preprocessing exit that allows you to make additional security checks or to instruct FRACHECK to accept or fail the request to access a resource. FRACHECK 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 security checking or to modify security 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 SVC. The PASSWORD and ALTUSER commands and the RACINIT SVC call this exit before actually changing the current password and/or password change interval.

RACF Encryption Exit: The ICHDEX01 exit allows you to control how RACF either encrypts (using a software implementation of the data encryption standard (DES) algorithm) or masks (using the RACF masking routine) the RACF password and operator identification card (OIDCARD) data that is stored in the RACF data set. The ICHDEX01 exit also allows you to entirely replace the encryption routine with whatever encryption routine the installation chooses.

For additional information on installation exits see *SPL: RACF*.

Tools for the security administrator

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

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 userid, the number and type of successful accesses, and the number and type of unauthorized access attempts.
- 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 the *Auditor's Guide*.

Using the Data Security Monitor

The data security monitor (DSMON) is a batch program that allows users with the AUDITOR attribute 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
- Program properties table report
- RACF authorized caller table report
- RACF exits report
- Selected user attribute report
- Selected user attribute summary report
- Selected data sets 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 these reports, see "Checking System Security" in Chapter 6 and/or the *Auditor's Guide*. The *Auditor's Guide* also contains details on use of the data security monitor.

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 the profiles. For DASD data sets, tape volumes, and other 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 RACHECK 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.

Listing Userids or Group Names Found in the RACF Data Set

You can list all occurrences of a userid or group name in the RACF data set, discover the relationships between various users and groups, and learn other important information about users, groups, and the resources they control by using the ICHUT100 utility program.

To invoke ICHUT100, you must be defined to RACF and must have the SPECIAL (or group-SPECIAL, as applicable) attribute. (If you do not have the SPECIAL or group-SPECIAL attribute, you can, however, list occurrences of your own userid.)

ICHUT100 produces a cross-reference report that describes the occurrences of each userid or group name you specify. Generic profile names are followed by the letter G in parentheses.

You can find complete information about the ICHUT100 utility and other RACF utilities in *SPL: RACF*.

Listing Information from RACF Profiles

The commands described in Figure 1-3 permit you and other authorized users to list the contents of profiles:

Command	Function
LISTDSD	Lists the contents of discrete or generic DATASET profiles. You can list the owner of the profile, the UACC designation, the date the profile was created, the users and groups authorized to access the data set(s), a count of the accesses to the data set or generic profile, and other information.
LISTGRP	Lists the contents of group profiles. You can list the owner of the group profile, the superior group name, the users connected to the group, the subgroup names, and other information.
LISTUSER	Lists the contents of user profiles. You can list 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, and other information.
RLIST	Lists the contents of discrete or generic profiles for general resources, such as tape volumes, DASD volumes, terminals, and IMS/VS transactions. You can list the owner of the resource, the date the resource was defined, the UACC designation, the users and groups authorized to access the volume, a count of accesses, and other information.
SEARCH	Obtains a list of user, group, and resource names from the RACF data set. (The search for resource names is based on a "mask," a character string that you specify with the command.) You can direct the output to a TSO CLIST data set.

Figure 1-3. Commands to List Profile Contents

The listings from the commands described in Figure 1-3 enable you to track and maintain control of all RACF-defined users, groups, and resources in your computing system.

Chapter 2. Organizing for RACF Implementation

The major intent of this publication is to describe 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. Among the topics briefly covered are:

- Ensuring management commitment
- Selecting and coordinating the security implementation team
- Defining installation security objectives and preparing an implementation plan
- Deciding what to protect
- Establishing ownership structures
- Educating the end users

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 divorce themselves from 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 supporting management position on 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. (And 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 Figure 2-1.

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. 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 RACF data sets. 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.
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.
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, you might select a data base administrator to represent protection for the DB/DC environment including: • DB/DC users • Accessibility to DB/DC subsystems • Terminal and transaction protection • Database protection for batch access

Figure 2-1. Participants of the Implementation Team

The remainder 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 will prepare 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 prioritize 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 will provide 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, according to the nature of its business. 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. Non-confidential 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.

Obviously, 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

You can protect existing DASD data sets with either discrete profiles or generic profiles.

Protecting New Data

RACF provides several ways to protect new data automatically:

- **Generic Profile Checking**: 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 data sets that have a similar naming structure. If your MVS system has the always call feature, generic profiles will protect existing data sets even if they are not RACF-indicated. (See "Protection Via Discrete Profile" in Chapter 4.)

- **ADSP or PROTECT**: New DASD data sets are automatically protected by the creation of a discrete profile when the user creating them has the ADSP

attribute, or when the PROTECT keyword is used on the JCL DD statement. This dynamic definition of data sets is accomplished when the resource manager issues the RACDEF macro instruction. Note: ADSP and the PROTECT keyword always causes the creation of a discrete profile and should normally be used only for data sets that have unique access-authorization requirements.

- **ADSP and Profile Modeling**: ADSP, by itself, allows only the creator of the data set to access the protected data. You can allow other people to access protected data by combining *profile modeling* with ADSP, in which case the created user or group data set automatically has an access list copied from the model.

  Profile modeling allows you to define a model profile for each user and/or group. Each model profile can contain defaults for the UACC, auditing flags, owner, level, installation-defined data, and access list.

  The MODEL operands of the ADDUSER, ADDGROUP, ALTUSER, and ALTGROUP commands allow you to automatically supplement the information normally placed in new RACF data set profiles by the ADSP, PROTECT, or ADDSD function. The modeling option pertains to user and group data sets, and is established by the SETROPTS command.

  You can also establish profile modeling using the RACDEF preprocessing exit routine.

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. Consider creating a resource profile that causes SMF logging to occur for all accesses of a resource, but which denies no accesses.

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, ALTDSD, or RALTER command that a WARNING is desired. Whenever RACHECK is about to fail a request for access, it will first check the resource profile to see whether the WARNING indicator is on. If it is, RACHECK will issue the warning message to the user; if not, RACHECK will deny the user's request.

Note: The warning message facility applies only to resource access checking via the RACHECK SVC. It does not permit any access via the RACF commands to the resource profile itself. In addition, the FRACHECK routine does not support the warning option.

Establishing Ownership Structures

RACF provides enough flexibility so that in most cases there should be no need for changes to the existing management and organizational structures. This does not mean, however, that some realignment of the existing 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 Userids 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 mold RACF to map into the current environment. For example, do any or all of the system users already have userids? If so, perhaps you can make use of them.

In TSO, for example, the existing userids tend to be ideal. Here, by default, all data set names have their owners' userids as their high-level qualifiers.

Other subsystems might also employ userids. If these existing userids are numeric, then exit routines can usually be prepared to prefix them with an alphabetic character to conform to RACF convention.

Batch Users: Batch users might not already have userids. Here, you might consider assigning userids based on personnel number or, if appropriate, userids based on group names. If it is not clear what to use as a userid, start by considering group names. Again, examine what already exists:

1. Is there an existing organizational structure that has groups with suitable abbreviations? Can the existing structure be used as is, or modified to suit?

2. What conventions already exist in job statements? It is common for the first few characters of the jobnames to be meaningful in terms of an application name, a project, a department, or some other such functional group. Could these be used as group names, or even a userid? Are there any other fields in the job statement that could be used (for example, account number or programmer name)? That is, could you look at a job statement and from it determine to whom or to which functional group the job belongs? (Note: The ability to derive a userid or group from existing job statement information can prove to be a significant migration aid. It could help you avoid the administrative effort to add the USER=keywords to existing job statements.)

3. Also look at data set names to examine the local data set naming conventions. Can you determine to which functional group a data set belongs by looking at the name? Can you say "This is an IMS data base," or "This data set belongs to the payroll group"?

   It is likely that several naming conventions exist. Using RACF options enables you to handle most existing variations.

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 userids and group names, while possible, is best avoided by careful initial selection.

You should map your groups to your organization's structure and should 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 2-2 shows relationships that can exist between users and groups.

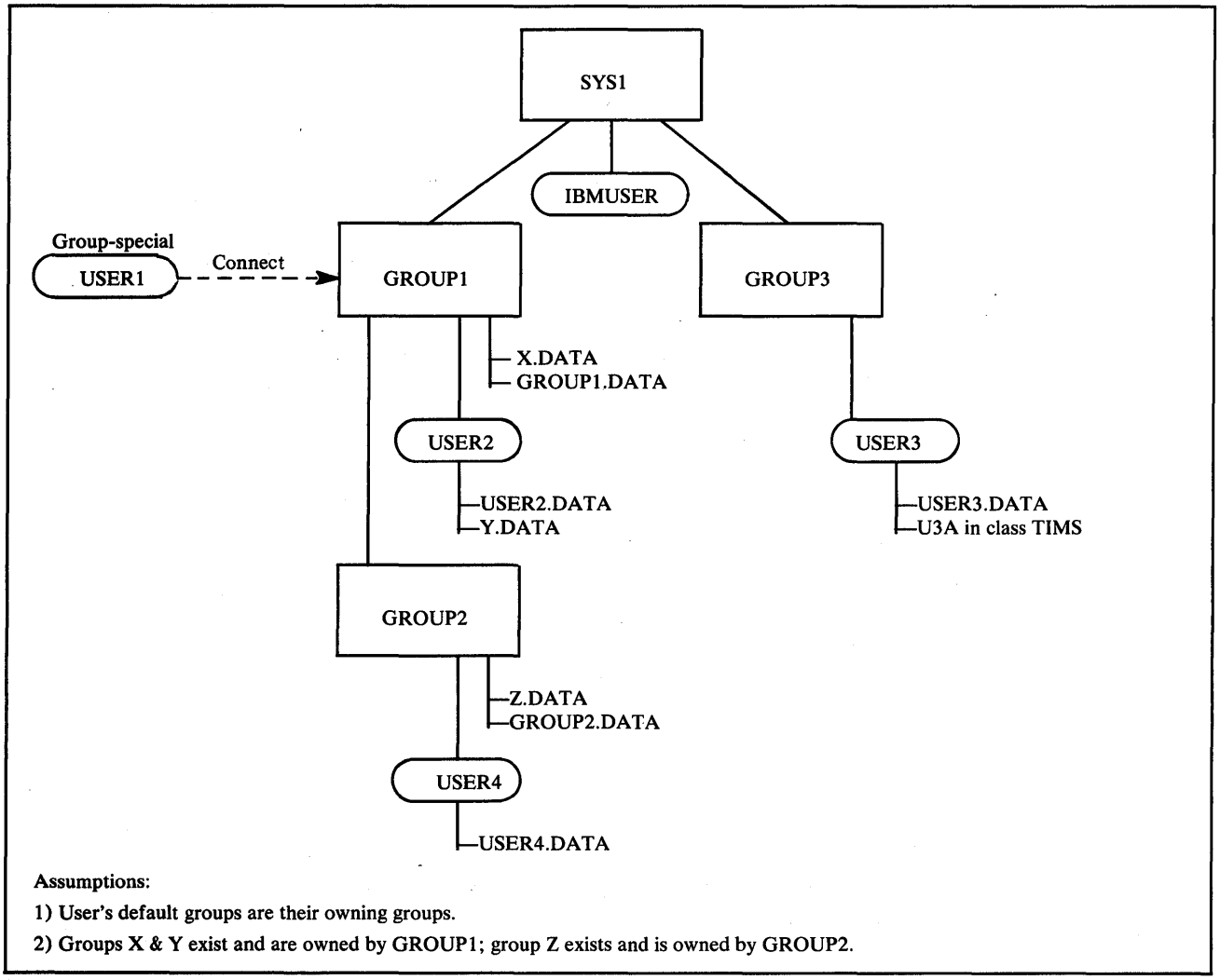


Figure 2-2. User and Group Relationships

In Figure 2-2, 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 (but not GROUP3's resources).

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 how to identify themselves to the system. This includes being able to use the RACF-related parameters on a JCL JOB statement and the TSO LOGON command. For information on RACF-related JCL and TSO parameters, see *JCL* and *TSO Command Language Reference.*

TSO Users and Administrators: TSO users and TSO administrators should be aware of the change in user password handling. (That is, the RACF-defined passwords are used for user verification instead of the passwords in the users' TSO UADS entry.) Therefore, is a user's TSO UADS entry contains more than one account number, the user must specify an account number on the LOGON command.

Users also need to know how to use the JOB statement and TSO LOGON command to change their passwords periodically. Users might need to be familiar with the RACF PASSWORD command if you want them to be able to change their password intervals. Users should be able to use the RACF LISTUSER command to list their own profile information.

IMS/VS and CICS/VS Administrators: IMS/VS and CICS/VS administrators should know how the RACF and IMS/VS or CICS/VS security functions interact with one another. They should understand the types of potential security problems that can be solved by using RACF in addition to the security features of their own subsystems. These considerations are discussed in Chapters 7 and 8 for IMS/VS and CICS/VS, respectively.

IMS/VS and CICS/VS Terminal operators: IMS/VS and CICS/VS terminal operators should be made aware of any changes to their sign on procedures required when RACF protection is implemented. These considerations are discussed in Chapter 7 for IMS/VS and Chapter 8 for CICS/VS.

Users Who Protect Data Sets with RACF: Users who protect data sets should be aware of necessary access requirements of potential users of these data sets. In addition, they should be aware of the benefits of selecting either generic or discrete profiles.

Users Who Maintain Protected Data Sets: Users who copy, move, or perform other similar actions on protected data sets must be aware of the requirements and implications of performing such operations. This information appears in *SPL: RACF.*

Users Who Maintain the RACF Data Set: Users who maintain the RACF data set (for example, technical support personnel) must be familiar with the RACF utilities. *SPL: RACF* describes these utilities.

RACF Command Users: Many of the commands and some of the operands on the commands are restricted to users who are owners of RACF entities (such as the owner of a group) or who have sufficient authority (such as JOIN group authority or ALTER access authority). You can limit the amount of information you give to a user to whatever is needed for the user to carry out the assigned level of responsibility. For example:

- Users responsible for administering user profiles, group profiles, data set profiles and other resources will need to know the RACF commands required to maintain and list the profiles. These include the ADDSD, ALTDSD, LISTDSD, PERMIT, DELDSD, RDEFINE, RALTER, RLIST, RDELETE, and SEARCH commands.

- Only users with the AUDITOR attribute need to know how to use the GLOBALAUDIT operand on the ALTDSD and RALTER commands and the UAUDIT operand on the ALTUSER command.

Note that if ISPF is installed, the requirements to know the various commands is greatly lessened because the command issuer is able to respond to the prompts on the ISPF entry panels.

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

Summary

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 resource 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. Figure 2-3 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.

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 naming conventions exist? Are changes necessary? Is implementing RACF to 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/Group Names	Establish names for groups and userids. 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 data sets can be protected by generic profiles and which data sets 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 exit routines are to be used, if any, and under what conditions (such as user derivation).
MVS Authorizations	Review MVS authorizations (the program properties table, APF libraries, and similar items).
Recovery	Establish recovery procedures.
Violation Procedures	Establish security procedures for logging, reporting, auditing.
Subsystems	Consider IMS, CICS and other subsystems.
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.
Install RACF	Select RACF options and install RACF.
Monitor	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.

Figure 2-3. Checklist for Implementation Team Activities

Chapter 3. Defining Groups and Users

This chapter provides in-depth information on defining groups and users. It includes information on:

- The types of RACF groups you might want to define
- Suggestions for defining RACF userids
- The various attributes you might assign and authorities you might grant a user
- The capabilities of a RACF owner
- The capabilities of a group-level user, according to the user's group authority
- Command limitations of the user possessing the group-SPECIAL or group-AUDITOR attributes
- And other information pertaining to group and user definitions

Defining RACF Groups

The group structure of RACF is capable of being 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 *data control* group, a *functional* group, or a *user* group?

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, who 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.

Data Control Groups: You can create a group to act as a control point for the protection of data. For example, the group "SYS1" is used to determine which users are permitted to protect the SYS1 data sets; only users with CREATE authority or higher in this group can protect system data sets. At your location, you or your delegate might consider defining one such group for every high-level qualifier representing data that is to be protected. Note: See also "Protection Via Generic Profiles" in Chapter 4.

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 data sets across many groups, such as accounting, payroll, marketing, and others. Of course, the owners of each data set could permit the financial analyst to access their data sets by placing the analyst's userid on an access list. But if a new financial analyst takes over the job, it is then necessary to add the new userid 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 data set 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, data set 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, access groups can be defined. For example, if PAY.DATA is a RACF-defined data set, two groups could be defined, PAYREAD and PAYUPDTE, both of which would appear in the PAY.DATA access list, but with READ and UPDATE access, respectively. Any users requiring access would be connected, as appropriate, by the group administrator.

Group Naming Conventions

The group naming conventions are relatively simple:

- A group name can be from one to eight letters, numbers, or combinations of the two. It must begin with a letter or an #, $, or @ sign.
- No two groups can have the same name. No group name can be the same as a userid.

As 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 one- or two-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 ensures that no two group administrators attempt to use the same group name.

Group Ownership and Levels of Group Authority

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

Ownership of a RACF Group

Each group you define to RACF must be owned by a RACF-defined user or by another RACF-defined group. You assign ownership of a group with the ADDGROUP or ALTGROUP command. 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: Being an owner does not, by itself, permit a user to access any user data sets or group data sets in the group; the owner, like other users, requires appropriate access authority to access data sets. However, as an owner the user does have authority to add himself to the access list of any profile he owns.

Group Ownership of Profiles

A *RACF group* can be assigned as the owner of a user, group, connect, data set, or 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 (see "Group-Related User Attributes" later in this section) will have the authority of SPECIAL for all profiles owned by the group 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 Figure 3-1.

Authority	Functions Permitted
USE	A user with the USE group authority can enter the system under control of that group, can access data sets to which the group is authorized, and can create RACF-protected user data sets.
CREATE	A user with the CREATE group authority can RACF-protect group data sets and control access to them. CREATE group authority includes the privileges of USE group authority.
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.
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.

Figure 3-1. Group Authorities

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 resource profiles.

  Another way to share administration responsibilities for the group, the users, and the group's resources is as follows: the owner of the group connects one user to the group with CREATE authority and all other users with USE authority. The owner of the group can then monitor the group and the users in the group; the user with CREATE authority can define and control the group resource profiles.

Group-Related User Attributes

A user to whom you give the SPECIAL, OPERATIONS, or AUDITOR attribute *at the group level* (via a connect profile) is limited to the resources within the scope of the group. That is, such a user does not have system-wide authority. However, if list-of-groups checking is active (see "Specifying List of Groups Authority Checking" for details) the user will automatically get the group-related authorities in all groups to which the user is connected, no matter which group the user logs on to.

See "User Attributes at the Group Level" for full details on specifying group-related user attributes.

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 this option (specified as NOTERMUACC), users of the group are authorized to log onto TSO from only 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 onto TSO from terminals (either RACF-defined or otherwise) based on the universal access authority of the terminals.

Defining Users

As a general objective, all users ought to be defined to RACF. However, RACF allows you to define as many or as few as you wish. Those users who are not defined to RACF can use the system virtually unimpeded, unless, of course, they attempt to access data to which they are unauthorized. (Access to protected data is allowed within the scope of the UACC assigned to the resource. For example, any user, defined or not, is able to review system libraries that have a UACC of READ.)

The users you must initially define are those you have selected for the pilot project and the central core of personnel who maintain and operate the system itself. Other users can then be defined as determined by convenience and the priority of their security needs.

There are some advantages in defining *all* users to RACF:

- Defining all users provides for better administrative control over who is using the system. This in turn can reduce misuse of system resources.
- Attempted violations by undefined users are difficult to investigate.

Whether all users are eventually defined to RACF is your decision; while not to be encouraged, you may deem individual accountability for a certain section of the user population unnecessary in some cases.

User Naming Conventions

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

- A userid must be from one to eight characters in length. It must begin with a letter or an #, $, or @ sign. (Note: although RACF permits 8-character userids, TSO users must keep in mind that TSO does not permit userids greater than seven characters.)
- No two userids can be the same. No userid can be the same as a group name.

Suggestions for Defining Userids

Basically, there are no requirements for establishing a specific type of userid. That is, in some installations, you might form userids 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 userid.)

The concept of userids based on group names appears practical because a quick glance at the userid 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 addition, how will such a user handle the "userid.data.sets" after the userid is changed? In the long run, userids based on something like personnel numbers do not have this problem and offer the greatest long term flexibility.

Where userids already exist in machine readable form (for example, in SYS1.UADS), a simple CLIST can provide a valuable administrative aid in migrating users to RACF.

Where userids are being assigned from scratch, they can often be created in blocks, again via CLIST. For example, you could centrally create 50 userids, 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 parameters can all be preset. You should assign the REVOKE attribute to unused userids.

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) 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).

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 user 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 attributes, described in the following seven topics, are:

- SPECIAL
- AUDITOR
- OPERATIONS
- CLAUTH
- GRPACC
- ADSP
- REVOKE

The SPECIAL Attribute

A user having the SPECIAL attribute at the user level can issue all RACF commands. The SPECIAL attribute gives the user full control over all RACF profiles in the RACF data set.

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 operator identification cards and passwords, and should change their passwords often to help ensure password security.

Note: Because all unprotected data sets are open to unauthorized revision, users having the SPECIAL attribute should take special care to protect their user data sets because they can contain sensitive information.

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" for additional details.

The AUDITOR Attribute

A user having the AUDITOR attribute at the user level has the authority to specify logging options on the ALTDSD, ALTUSER, RALTER, and SETROPTS commands. In addition, the auditor can list auditing information with the RLIST, LISTDSD, LISTUSER, LISTGRP, and SEARCH commands, as well as with the ICHUT100 utility program. The AUDITOR attribute gives the auditor control of logging to the SMF data set to help detect changes (or attempted changes) to the RACF data set 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 data set; nor does it give him the authority to access protected data.

A user having the AUDITOR attribute can run the data security monitor (DSMON) program to produce reports that are useful in monitoring the status of the installation's data security environment. The auditor can compare the information in the reports with the intended resource security controls at your installation and determine any discrepancies between the actual and expected results. (For more information on the data security monitor, see the *Auditor's Guide.*)

You should assign the AUDITOR attribute to only those 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.

Note: Because all unprotected data sets are open to unauthorized revision, users having the AUDITOR attribute should take special care to protect their user data sets because they can contain sensitive information.

You can assign the AUDITOR attribute at the group level. When you do, the *group-AUDITOR* user's authority is limited to profiles owned by that group only. See "User Attributes at the Group Level"for detailed information.

The OPERATIONS Attribute

A user having the OPERATIONS attribute at the user level (unless specifically excepted by an entry in the resource profile) has full authorization to all RACF-protected data sets and resources defined in the class descriptor table. In addition, the OPERATIONS user has full control over data set profiles that control group data sets. The user's authorization includes the ability to: copy, reorganize, catalog, and scratch data sets; perform input/output operations on tape volumes; create or destroy labels on tape volumes through OPEN and end-of-volume operations; and perform all online DFDSS functions.

A user having the OPERATIONS attribute can define and modify the profiles for group data sets (for groups to which the user is not connected).

The OPERATIONS authority can be limited for specific resources if the OPERATIONS user is specifically authorized to the resource with a lesser authority (such as READ) via the PERMIT command. In this case, the OPERATIONS user's assigned level of access authority takes precedence over the OPERATIONS attribute.

You should assign the OPERATIONS attribute to the minimum number of personnel.

To reduce the number of users having the OPERATIONS attribute at the user level (and therefore have the attribute system-wide), 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. See "User Attributes at the Group Level" for detailed information.

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

The CLAUTH (Class Authority) Attribute

A user to whom you assign the CLAUTH attribute is authorized to define entities to RACF in the USER class or classes defined in the class descriptor table. The user to whom you assign the CLAUTH attribute is authorized to define new users to RACF with the ADDUSER command (USER class), provided the user is the owner or has JOIN authority in the new user's default group.

The user to whom you assign the CLAUTH attribute is also authorized to define (with the RDEFINE command) resources of the types defined in the class descriptor table. Valid class names are: TAPEVOL, DASDVOL, TERMINAL, TIMS (IMS/VS transactions), GIMS (IMS/VS transaction groups), APPL (applications), AIMS (IMS/VS application group names), TCICSTRN (CICS/VS transactions), GCICSTRN (CICS/VS transaction groups), PCICSPSB (CICS/VS program specification blocks, or PSBs), QCICSPSB (CICS/VS PSB groups), and DSNR (DB2), plus any installation-defined class names.

You should give the CLAUTH attribute to only those users who are responsible for defining entities in these classes to RACF.

The CLAUTH attribute can be delegated only by a user (security or group administrator) having the SPECIAL attribute, or by a user who already has the CLAUTH attribute for the class authority being delegated.

The GRPACC (Group Access) Attribute

If a user has the GRPACC attribute, any group data set profiles he defines to RACF (via the ADSP attribute, PROTECT parameter on the DD statement, or ADDSD command) are automatically made accessible to other users in the group. (The group whose name is used as the high-level qualifier of the data set name is given UPDATE authority to the data set.)

Note that, if the defining user does not have the GRPACC attribute, the user must use the PERMIT command to allow the group to access the group data set.

You should assign the GRPACC attribute with care, especially if the RACF user to whom you are assigning the attribute is allowed to RACF-protect group data sets in several groups; this user could unintentionally authorize groups to access a group data set to which they should not have access.

Only the owner of a user's profile or a user with the SPECIAL attribute can assign or unassign the GRPACC attribute.

The ADSP (Automatic Data Set Protection) Attribute

When you assign the ADSP attribute to a user, any permanent data sets that the user creates are automatically defined to RACF with discrete profiles. This attribute is extremely useful in the initial phase of RACF operation to quickly establish protection of new data sets without educating or otherwise involving the end user in the process.

When a user has the ADSP attribute, the RACDEF SVC always automatically creates a discrete profile. Therefore, if generic profile checking is active and data management always call is installed at your location, you should normally revoke the user's ADSP attribute. You can do this on a user-by-user basis, or for an entire installation with the NOADSP parameter of the SETROPTS command.

A data set created under ADSP is accessible only to the user who created it, unless the GRPACC user attribute and/or modeling is also used. Only the owner of a user's profile (or a user with the SPECIAL attribute) has control over the ADSP attribute.

Caution: A DASD data set is defined to RACF at allocation. If the data set disposition is altered at deallocation (through dynamic deallocation), the change is **not** reflected in the RACF data set. For example, if the data set disposition is DELETE at allocation and KEEP at deallocation, the data set is not automatically RACF-protected.

The REVOKE Attribute

You can prevent a RACF user from entering the system by assigning the REVOKE attribute. 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.

Only the owner of a user's profile (or a user with the SPECIAL) attribute) has control over the REVOKE attribute.

User Attributes at the Group Level

You can specify the SPECIAL, AUDITOR, OPERATIONS, GRPACC, and REVOKE 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, group-OPERATIONS, and so on, to distinguish them from attributes at the user level.

Group attributes are indicated in the connect profile, and are in effect for the user only when the user is connected to the group during a batch job or terminal session. (However, when list-of groups checking is in effect, the group-SPECIAL, group-OPERATIONS, and group-AUDITOR user automatically has group-related authorities in all groups to which the user is connected, regardless of the group the user is logged on to.)

When you initially define a new user, the user's connect profile to the default group has no group-related attributes indicated. You can use the CONNECT command after you initially define the user to modify the user's connect profile to 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:

- Resources owned by the group
- Resources owned by users who are owned by the group
- Resources owned by subgroups that are owned by the group
- Resources owned by subgroups owned by subgroups, owned by the group, and so on.

Note that the scope of the group does not extend to resource profiles that are owned by groups that are owned by **users** who are owned by the group. Neither does the scope of the group extend to resources that are owned by users who are owned by **users** who are owned by the group.

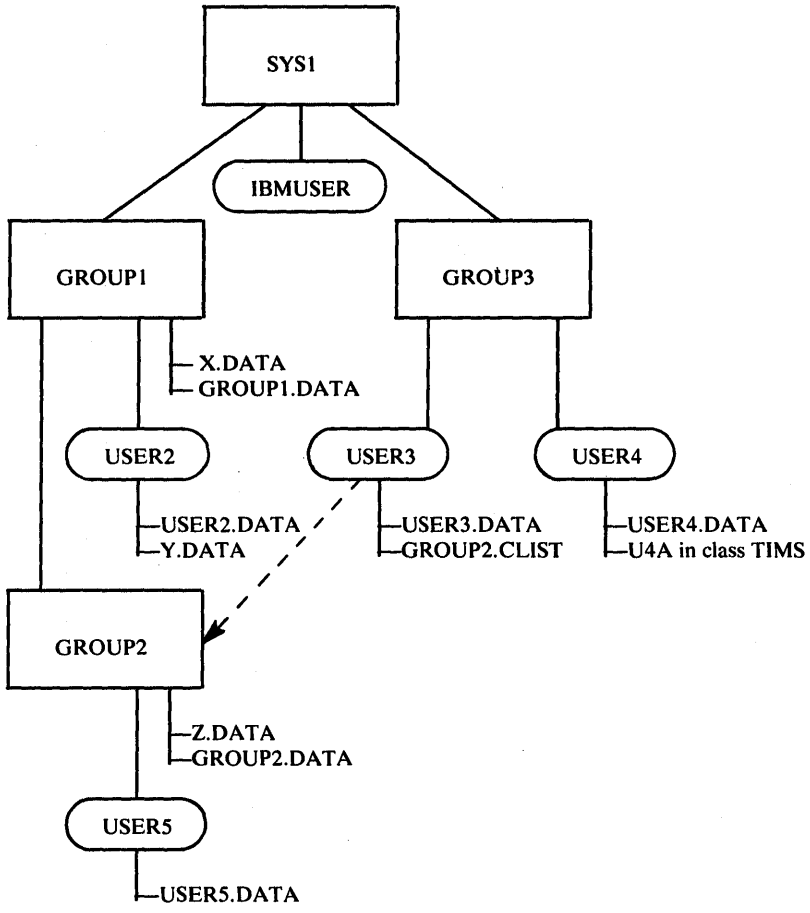
By establishing the group structure so that subgroups are owned by their superior groups, the authority of the group-SPECIAL, group-OPERATIONS, and group-AUDITOR user can be made to percolate down through the group tree structure as far as the security administrator desires. When a user's attribute percolates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

The limits of the security administrator, group administrator, auditor, and operations personnel authority at the group level are described in Figure 3-2. (Of course, these users continue to have whatever authorities they possess from other sources, such as ownership and list membership, that are not covered by their group level authorities.)

Resource	Attribute, User, and Authority
Data Sets	**Group-SPECIAL Attribute**: A user with the group-SPECIAL attribute has full authority to access: • Data set profiles that are owned by the group • Data set profiles having a high-level qualifier equal to the group identifier • Data set profiles owned by users or groups that are owned by the group • Data set profiles having a high-level qualifier that is a user or group identifier owned by the group The group-SPECIAL user can also define data set profiles with a high-level qualifier that is the group identifier or a user or group identifier 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 data sets as the user with the group-SPECIAL attribute.
General Resources	**Group-SPECIAL Attribute**: A user having the group-SPECIAL attribute has full authority to access: • 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 having 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.
Users	**Group-SPECIAL Attribute**: A user with the group-SPECIAL attribute has full authority to access: • User profiles of users owned by the group • User profiles of users 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 user 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 having 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 having 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.

Figure 3-2. Scope of Authority for User Attributes at the Group Level

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



Assumptions:

1) User's default groups are their owning groups, except for USER5 who is owned by another user in GROUP2.

2) Groups X & Y exist and are owned by GROUP1; group Z exists and is owned by GROUP2.

Figure 3-3. Group Level Authority Structure

Figure 3-4 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 in Figure 3-4.

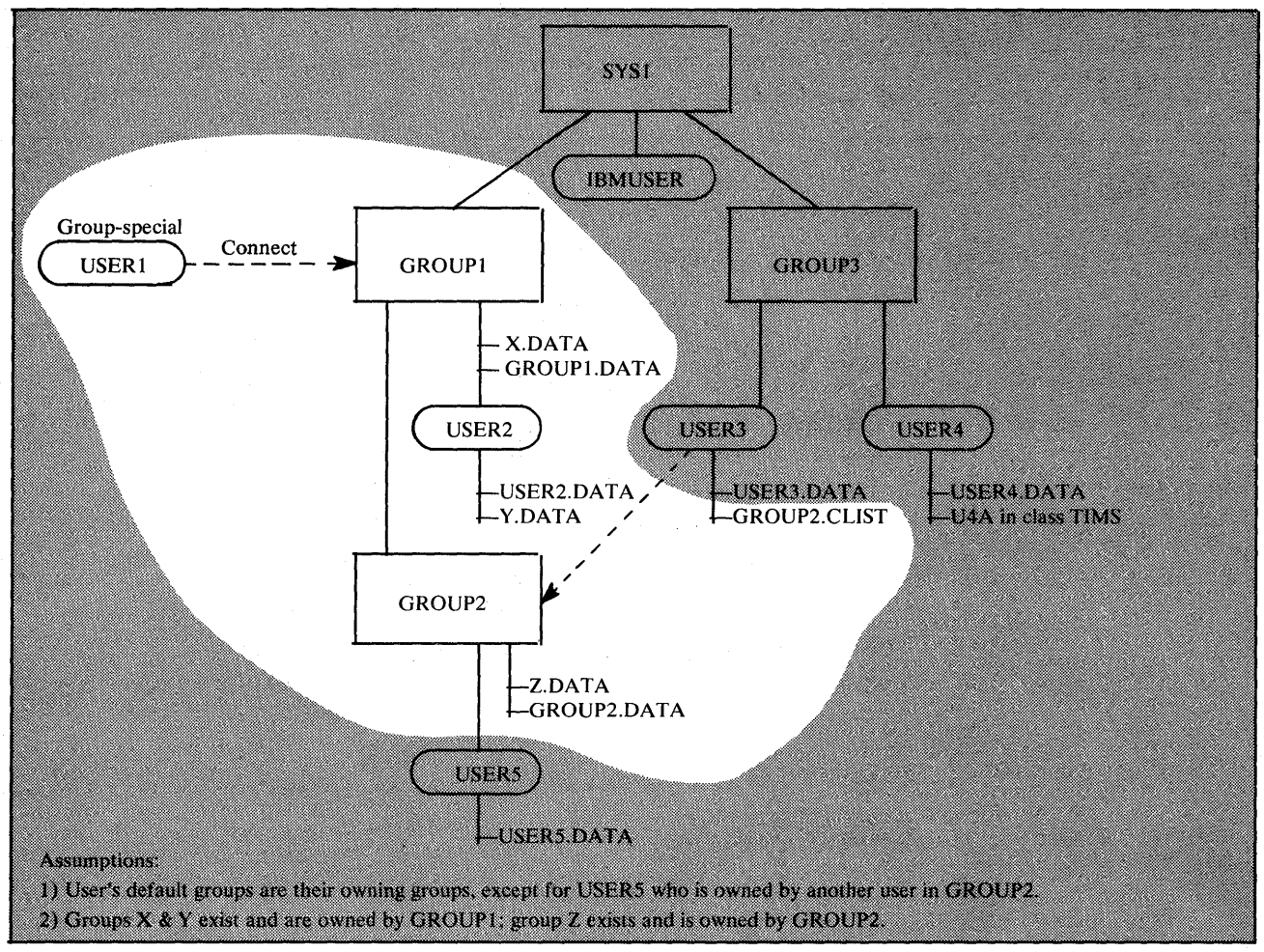


Figure 3-4. Scope of Authority of a Group-SPECIAL user

In Figure 3-4, USER1 has authority to the indicated resource profiles for the reasons listed in Figure 3-2. USER1 does not have authority to any of the resources in the shaded area for the following reasons:

- GROUP3 is not owned by GROUP1.
- USER3 is not owned by GROUP1.
- USER4 is not owned by GROUP1.
- USER5 is not owned by GROUP1 or GROUP2.
- USER3.DATA is not a data set owned by a user who is owned by GROUP1.
- USER4.DATA is not a data set owned by a user who is owned by GROUP1. 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 ICHUT100, USER1 would be informed 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.

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.

Note that you cannot assign the ADSP attribute to a user who allocates space for data sets that do not meet the RACF or installation naming conventions.

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 who 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 in a group is assigned a default universal access authority (UACC) of NONE, READ, UPDATE, CONTROL, or ALTER. The value of NONE is used as a user's default universal access authority unless you specifically assign the user another value on the ADDUSER, ALTUSER, or CONNECT command. This value is always assigned (unless modeling is used) as the universal access authority to a data set profile that is automatically defined when a user who has the ADSP attribute allocates a new DASD data set, and to a DASD data set profile or tape volume profile when the PROTECT parameter is specified on the JCL DD statement. It is also in effect when the user RACF-protects a resource with the ADDSD or RDEFINE command and does not specify a value in the UACC operand.

Chapter 4. Defining Resources

This chapter contains in-depth information on defining resources to be protected. Among the major topics included are:

- Protecting DASD data sets
- Protecting DASD volumes
- Protecting tape volumes
- Protecting terminals
- Other general resource considerations

4

Protecting DASD Data Sets

This section describes considerations related to using RACF to protect DASD data sets.

Rules for Defining Data Set Profiles

When you define data set profiles to RACF, you can choose to use standard or non-standard naming conventions, as described under the following two topics. The descriptions of standard and non-standard conventions are followed by rules for protecting the three types of data sets: user, group, and other.

Standard Data Set Naming Conventions

By default, RACF expects the high-level qualifier of the name of a data set profile to be either a RACF-defined userid or group name.

If you and your implementation team have chosen to define data set profiles under the standard RACF naming conventions, you can create groups for each high-level qualifier and permit authorized users to protect any data set having that high-level qualifier by giving them CREATE authority in that group.

As a secondary benefit of using the standard RACF naming conventions, you can automatically enforce naming conventions at your location. That is, when your users have the ADSP attribute, they can neither protect nor allocate data sets other than those for which they have CREATE or higher authority (except for data sets whose names begin with their own userid).

If your location has installed data management always call, you can use a RACDEF exit routine to require that a predefined generic profile exists before a data set allocation can be done. Effectively, this enforces the use of a predefined naming convention.

Non-Standard Data Set Naming Conventions

One of the nice features of RACF is that you can employ options to modify existing data set names to make them conform to RACF standard naming conventions. This means that your users, if you so decide, can continue to use data set names that existed before installing RACF, even if the names do not conform to RACF naming conventions.

RACF accepts such names but modifies them internally in order to protect the data sets. For example, the single-level name prefixing facility of RACF adds a qualifier to make the data set name acceptable to RACF routines. In this case, all SMF log records and messages from RACF contain the RACF-modified version of the data set name.

Still, the data set name originally defined by the user is the name the user (or the group administrator or auditor) would probably prefer to see on log printouts and messages. RACF offers an option that causes RACF to use the original, user-defined data set names, called "real data set names," when printing reports or issuing messages. See "Specifying Real Data Set Names" in Chapter 5 for a detailed discussion of this option.

Note: This option has no effect on single-level data set names whose "real data set names" continue to be the prefixed ones; this option applies only to name conversions made by the naming conventions table, or by installation exit routines.

Table-Driven Data Set Naming Conventions

You have the ability to create a naming conventions table (ICHNCV00), which will be used to check the data set name in all commands and SVCs that process data set names. Creating this table will help you set up and enforce data set naming conventions that are different from the standard RACF naming convention. That is, the table can selectively rearrange data set names to "fit" the RACF convention without actually changing those names. (Because the table-driven naming convention processing is done before other preprocessing or naming convention exits are called, the existing exits can still be used for additional processing.)

Protecting User Data Sets

A **user data set** is a data set whose high-level qualifier is a RACF userid. The following rules apply to user data sets:

- All RACF-defined users can protect their own data sets.
- A user can RACF-protect a data set for another user by means of a discrete or generic profile under any of the following conditions:
  - The RACDEF preprocessing exit routine allows RACF protection.
  - The user who is protecting the data set has the SPECIAL attribute (or the group-SPECIAL attribute in the group that owns the user profile), and the request is made using the ADDSD command.

Protecting Group Data Sets

A **group data set** is a data set whose high-level qualifier is a RACF group name. A RACF-defined user can RACF-protect a group data set under any of the following conditions:

- The user has the SPECIAL attribute (or the group-SPECIAL attribute for that group) and the request is made using the ADDSD command.
- The user has JOIN, CONNECT, or CREATE authority in the group.
- The RACDEF preprocessing exit routine is used to override normal RACF authorization requirements.

Controlling the Allocation of New DASD Data Sets

The following cases describe how RACF can be used to control the allocation of new user and group DASD data sets.

A user can allocate a **user** data set in the following situations:

- The system has always call, the data set name is protected by an existing generic profile, and the user does not have ADSP.

  RACF allows the allocation if (1) the user has ALTER authority to the data set via the generic profile or global access checking, or (2) the data set is the user's own data set. RACF does not create a profile.

- The data set name is not covered by an existing generic profile and the user does not have ADSP.

  RACF allows the allocation, but does not create a profile.

- The user has ADSP and the data set is the user's own data set.

  RACF allows the allocation and creates a discrete profile for the data set.

A user can allocate a **group** data set in the following situations:

- The system has always call, the data set name is protected by an existing generic profile, and the user does not have ADSP.

  RACF allows the allocation if at least one of the following is true:

  - The user has ALTER authority to the data set via the generic profile or global access checking.
  - The user has CREATE authority in the group.

  RACF does not create a profile.

- The data set name is not covered by an existing generic profile and the user does not have ADSP.

  RACF allows the allocation, but does not create a profile.

- The user has ADSP and the data set belongs to a group of which the user is a member.

  RACF allows the allocation only if the user has CREATE authority in the group. If RACF allows the allocation, it creates a discrete profile for the data set.

Any user without ADSP can allocate a data set whose high-level qualifier is neither a RACF userid (user data set) nor a RACF group name (group data set), but the data set cannot be RACF-protected. Note that a dummy group (a group that has no users connected to it) can be defined for the high-level qualifier of these data sets so that they can then be RACF-protected.

Note: In all cases, if the user specifies the JCL parameter PROTECT=YES or the TSO ALLOCATE parameter PROTECT (these parameters request that RACF create a discrete profile), RACF treats the user the same as a user with ADSP. However, because the use of these parameters is voluntary, an installation cannot use the parameters to control allocations.

Ownership and Access Authorities for DASD Data Sets

This section describes the ownership of a RACF-protected DASD data set, the access authorities, and suggestions for assigning access authorities.

Data Set Profile Ownership

Each DASD data set profile defined to RACF requires a RACF-defined user or group as the owner of the profile. The owner (if a user) has full control of the profile, including the access list for the data set(s) that the profile applies to. Ownership of data set profiles is assigned when the profiles are defined to RACF. Note that ownership of a data set profile does not mean that the owner can automatically access that data set. In order to access a data set, the owner must still be authorized in the profile's access list.

Access Authorities for DASD Data Sets

You permit users and groups to access a RACF-protected DASD data set by adding them to the access list of the discrete or generic profile that applies to the data set. When you permit other users and groups to access a RACF-protected DASD data set, you give them one of the following access authorities.

ALTER Access Authority: Users and groups with ALTER access authority in discrete data set profiles have full authority over the profile, including the access list; in addition, they have total control over the data set protected by the profile, including the ability to scratch and rename the data set. Users and groups with ALTER access authority in generic data set profiles have total control over the data set(s) protected by the profile, including the ability to scratch and rename the data set(s), but they have no control over the profile or its access list. In an always call system, a user or group who has ALTER authority in the access list of a generic profile can allocate new data sets that will be covered by that profile.

For a user data set profile, the user whose userid matches the high-level qualifier of the data set profile automatically has the equivalent of ALTER access authority (even though the user might not be in the access list).

CONTROL Access Authority: For VSAM data sets, users and groups with CONTROL authority have an access authority to data sets that is equivalent to the VSAM CONTROL password.

For non-VSAM data sets, CONTROL authority is equivalent to UPDATE authority.

UPDATE Access Authority: Users and groups with UPDATE access authority can open a data set to read and write to the data set.

READ Access Authority: Users and groups with READ access authority can open a data set only to read the data set.

NONE Access Authority: Users and groups with NONE access authority cannot access the data set.

Universal Access Authority (UACC) for DASD Data Sets: Each DASD data set profile you define with RACF requires a universal access authority, which specifies the authority that users and groups who are not in the profile's access list have to the data set(s) that the profile applies to. Coverage by the UACC also includes users who are not defined to RACF and RACF users who submit batch jobs without identifying themselves as RACF users on the JCL JOB statement.

If you specifically assign to a user or group an access authority for a DASD data set (via the profile that applies to the data set), this specified authority overrides the UACC assigned to the data set. For a given data set:

- If you set UACC to NONE, all accessors not specifically authorized to access the data set are refused access to the data set.
- If you set UACC to READ, UPDATE, CONTROL, or ALTER, all users can access the data set at the specified level of authority, unless they are specifically excluded by their entry in the access list.

Suggestions for Assigning Access Authorities to DASD Data Sets

When protecting catalogs and CVOLs (control volumes), be sure that users and groups have a sufficient level of access authority to each protected entity along the path to a data set they are required to access.

The level of RACF access authority that a user or group requires to perform operations on VSAM data sets or catalogs is similar to the level of authorization required when passwords are used. (See "Comparison of Password and RACF Authorization Requirements for VSAM" later in this chapter.)

For a discussion of the levels of RACF access authority required to perform operations against OS CVOLs, see *OS/VS2 MVS CVOL Processor* or *MVS/XA Catalog Users Guide*.

Note that, when a discrete DASD data set profile is created through the RACDEF SVC, the installation exit routine can specify the universal access authority and an access list directly or can name an existing profile to be used as a model.

Data Set Profiles

The next three topics describe what occurs when data sets are protected by profiles.

Protection Via Discrete Profile

Users can protect DASD data sets with discrete profiles in the following ways:

- Automatically when they allocate a permanent DASD data set, if they have the ADSP attribute and ADSP is active on the system
- When they specify the PROTECT parameter on a JCL DD statement or on the TSO/E ALLOCATE command for new, permanent, DASD data sets
- When they issue the ADDSD command with the SET keyword for permanent, existing, online data sets

Two steps occur when you define a data set with a discrete profile. Only when RACF has completed both steps is the DASD data set protected:

1. An indicator is set to notify the system that the data set is RACF-protected with a discrete profile. The indicator is in the DSCB for a non-VSAM data set or in the catalog entry for a VSAM data set. This condition is called *RACF-indicated.* Note: For information on moving RACF-indicated data sets to other systems, and for information on using utilities with RACF-protected data sets, see *SPL: RACF.*

2. The discrete profile is added to the RACF data set. The profile contains the characteristics of the data set and a list of those users (the *access list*) authorized to access the data set.

   Note: Scratching a data set that is RACF-protected with a discrete profile causes RACF to delete the data set profile from the RACF data set.

Protection Via Generic Profiles

Using generic profiles means that your installation can reduce both the number of required profiles and the size of the RACF data set, thus making RACF protection easier to administer. In addition, generic profiles are loaded into storage when first needed, are not deleted when the data set they control is deleted, and are not volume-specific (that is, data sets controlled by a generic profile can reside on any DASD volume).

You can define a generic profile by using the GENERIC keyword with the ADDSD command. The generic profile name can contain one or more generic characters (% or *); however, a generic profile name need not contain any generic characters (in which case it looks just like a discrete profile).

Rules for the specification of generic characters follow.

Rules for Generic Profiles in the data set Class:

- Valid generic characters are % and *. The % in the generic profile name indicates that any single character in the same position of a data set name is a match to that character position in the generic profile. The * in the generic profile name indicates that any character in that position of a data set name and all subsequent characters (in that data set qualifier) are a match.
- The high-level qualifier of a generic data set profile name must not be, nor may it contain, a generic character. Examples:

ABC.EF*	Valid
ABC.E%*	Valid
A%C.EFG	Invalid
*.EFG	Invalid
ABC*.XYZ	Invalid

- Each qualifier in a generic profile name must have at least one character (generic or otherwise), and can be as long as eight characters. The last qualifier can be nine characters long; however, the ninth character, if included, must be an asterisk. (The addition of the ninth character makes it possible to use one generic profile to protect both a GDG basename and all of its generations.) Example:

Generic Profile Name	Protected GDG Names
GDG.BASENAME*	GDG.BASENAME GDG.BASENAME.G0123V00

- Combinations of the generic characters can be included in a single qualifier. Example:

Generic Profile Name	Data Set Names That Match	Data Set Name That Does Not Match
ABC.%%F*.*.*	ABC.DEF.G.HIJK ABC.LMFGH.NOPQ.RST	ABC.DEG.FHI.G

- If the last qualifier of the generic profile name contains an *, data set names having any number of subsequent qualifiers match the generic profile name. Example:

Generic Profile Name	Matching Data Set Name
ABC.EFG.*	ABC.EFG.HIJKL ABC.EFG.H.IJKLM ABC.EFG.HIJ.KLM.NOP

Generic Profile Checking for the DATASET CLASS: During access-authorization checking, RACF examines 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 generic profile. You should also keep in mind that RACF maintains discrete and generic profiles in internal-sort order, regardless of the order in which the profiles are created. Figure 4-1 lists some discrete and generic profiles from the DATASET class. This figure is in internal-sort order, and represents the order in which RACF checks the profiles when it performs access-authorization checking. That is, RACF first performs authorization checking for DASD data sets based on the high-level qualifier of the data set name. (In the case of single-level data set names, RACF optionally prefixes the single-level name with an installation-specified userid or group name.) The high-level qualifier must be a RACF-defined userid or a group name.

Profile Name Assigned By Security Administrator	Profile Type
SALES.DATA	Fully-Qualified Generic
SALES.DISK.APRIL	Discrete
SALES.DISK.*	Generic
SALES.%ATA	Generic
SALES.*.QUOTA	Generic
SALES.*.QUOT%*	Generic

Figure 4-1. Profile Names and Types

If data management always call is installed at your location, RACF is invoked whenever a data set is accessed (whether or not the data set is RACF-indicated) and whenever DASD space is allocated for a data set (whether or not the user has the ADSP attribute or has specified PROTECT=YES on the JCL statement). When RACF is invoked because of always call (and not because of RACF-indication), RACF checks only predefined generic profiles. When RACF cannot find an appropriate generic profile, it accepts the access request by default.

Caution: Data sets that are not RACF-indicated, but are protected by generic profile and always call are NOT PROTECTED if they are transferred (in any way) to another system that does not have RACF, always call, and appropriate predefined generic profiles.

Generic data set protection is less comprehensive when RACF is installed on systems without always call. That is:

- RACF security checking is invoked only for those data sets that are RACF-indicated.
- When access to a data set is requested, RACF first searches for a discrete profile. If no discrete profile is found, the most specific generic profile is used. If no generic profile is found, the access request is denied.
- Existing data sets can be made RACF-indicated by using the ADDSD command, followed by the DELDSD command with the NOSET keyword.

Note: Always call is not supported by current data management offerings. However, IBM has issued a statement that it is the intent of IBM to have Hierarchical Storage Manager and future data management products that interface with RACF Version 1 Release 5 and subsequent RACF releases support the always call interface.

Authority to Modify Generic Profiles

To modify a generic profile, a user must be the profile owner, or have the SPECIAL (or group-SPECIAL, if applicable) attribute, or have a userid identical to the profile's high-level qualifier. Unless one of these conditions is met, the user cannot alter the generic profile, even though the user may have ALTER access authority to the data sets that the generic profile protects. (See the *RACF Command Language Reference* for a description of user authorization.) Note that the access list in the generic profile does not apply to the profile itself.

Password-Protected Data Sets

When a DASD data set is both password-protected and RACF-protected, access to the data set is authorized through RACF authorization checking. If an authorization request for a password-protected data set is satisfied by a RACF global access table entry or a RACF data set profile, password checking is ignored.

When a DASD data set is password-protected but not RACF-protected, access to the data set is authorized through password protection.

When a RACF-protected DASD data set is moved to a system without RACF support, RACF authorization checking cannot be done. (See *SPL: RACF* for more details.) Therefore, once you have installed RACF, your users may need to maintain password protection only for those data sets that:

- Are not RACF-protected
- Are RACF-protected and are used on other systems that do not have RACF support

Comparison of Password and RACF Authorization Requirements for VSAM

The password authorization required for operations on VSAM data sets and data sets cataloged in VSAM catalogs is described in *Access Method Services.*

The RACF authorization requirements are the same as the password requirements for most VSAM operations. The RACF authorization levels of ALTER, CONTROL, UPDATE, and READ correspond to the password levels of MASTER, CONTROL, UPDATE, and READ.

As an example, deleting a VSAM data set requires the MASTER-level password of either the data set or the catalog that describes the data set. To delete a RACF-protected VSAM data set requires ALTER authorization to the data set or the catalog.

There are a few exceptions to the one-to-one correspondence of the RACF and password authorization levels. Figure 4-2 shows the exceptions to the one-to-one correspondence of RACF and password authorization requirements. The first column lists the VSAM catalog operations that have different RACF and password authorization requirements. The second column shows the password authorization required if passwords are used without RACF. The third column shows the non-corresponding RACF authorization required if both the catalog and the data set are RACF-protected.

VSAM Catalog Operation	Password Authorization Required	RACF Authorization Required
ALTER non-VSAM data set including RECATALOG)	Update to catalog	UPDATE to non-VSAM data set or ALTER to catalog
ALTER/DELETE GDG base	Update to catalog	ALTER to GDG base or ALTER to catalog
DELETE non-VSAM data set (UNCATALOG)	Update to catalog	ALTER to non-VSAM data set or ALTER to catalog
DELETE alias	Update to catalog	ALTER to real data set or ALTER to catalog
Export disconnect user catalog	Update to master Catalog	ALTER to user catalog or ALTER to master catalog
Define new GDG generation data set (CATALOG a GDG generation)	Update to catalog	UPDATE to catalog plus UPDATE to GDG base or ALTER to catalog

Figure 4-2. Variant Password and RACF Authorization Requirements

Protecting DASD GDG Data Sets

RACF protects GDG data sets residing on direct access volumes in one of three ways:

1. It allows the definition of a generic profile to protect all members of a generation data group. This method is identical to that for non-GDG data sets protected by a generic profile. Example: a profile of the form 'GDG.BASENAME*' protects all members of a GDG and the base entry for the GDG in the catalog.

2. It protects GDG data sets in the same way it protects non-GDG data sets that have discrete profiles.

3. It allows all members of a generation data group to use a single RACF data set profile. By specifying the MODEL(GDG) keyword on the SETROPTS command, the owner of the generation data set can establish a base (index)

name profile containing an access list that is accessible by all related users and groups. If you want individual access lists, do not create the profile for the base name. Then RACF processing will be the same as that performed for non-GDG data sets.

Note: The base name profile should not be a model type profile. It should be a data set profile containing the volume serial number (VOLSER) of the CVOL that the GDG is cataloged on. For more information on GDGs cataloged in CVOLs, see *OS/VS2 MVS CVOL Processor* or *MVS/XA Catalog Users Guide*.

Protecting VSAM Data Sets With VSAMDSET Group

Use the RACF-defined VSAMDSET group to protect VSAM data sets that are created with data set names that have VSAMDSET as their high-level qualifiers. (When RACF is installed, RACF sets a "universal group authority" of CREATE in the VSAMDSET group profile.) Because the users are not connected to the group, they do not automatically have access to the VSAMDSET group data sets.

This feature simplifies the usability of the VSAMDSET group and reduces the number of users who have access to the VSAMDSET group data sets.

Note: RACF users who are connected to the VSAMDSET group can define, and are allowed access to, VSAMDSET group data sets in the normal manner.

Protecting Data Sets That Have Single-Level Data Set Names

Installations that have data sets with names consisting of a single qualifier can still RACF-protect those data sets by using the SETROPTS command to define a prefix (high-level qualifier) for internal use by RACF when it processes requests for the single-level names. The prefix should be an existing group name, and cannot be the name used as the high-level qualifier of any actual data sets in the system. All RACF commands and the RACF report writer expect to see the prefix followed by a period and the single-level data set name in subsequent references to the profile. See Chapter 5 for more detailed information on how to specify protection for data sets that have single-level names.

Protecting Data Sets That Have Duplicate Names

You can use separate, discrete profiles to define DASD data sets having the same name. RACF differentiates between data sets having the same name by examining the volume serial number of each separately-protected data set.

Non-VSAM Data Sets: For non-VSAM data sets, RACF uses the serial number of the volume on which the data set resides.

VSAM Data Sets: For VSAM data sets, RACF uses the volume serial number of the VSAM catalog in which the data set is cataloged.

DFEF Catalogs: Multiple Data Facility Extended Function (DFEF) (5740-XYQ) catalogs, each of which contains a duplicate data set name, cannot reside on the same volume.

Support for data sets with duplicate names allows authorized users to:

- Move and copy RACF-protected data sets from one volume to another (for example, with the IEHMOVE system utility)
- Establish separate discrete profiles (including the access list and statistics and logging options) for data sets having the same name
- Protect data sets that have the same name and reside on non-shared volumes (such as SYS1.LINKLIB) on a loosely-coupled system that uses a shared RACF data set

Preventing RACF Protection of Data Sets That Have Duplicate Names

You can prevent identically-named DASD data sets from being defined to RACF with separate, discrete profiles by modifying the installation-replaceable module ICHSECOP. (See "Disallowing Duplicate Data Set Names" in Chapter 5 for information on selecting this option.) You can use this option to allow data sets with the same name to be defined to RACF for protection in one common profile. In this case, a data set shares the data set profile (including the access list, and statistics and logging options) with other data sets that have the same name.

Note that a generic profile (with a fully-qualified name) can also be used to protect data sets with identical names, regardless of what volumes they reside on.

Using the PROTECT Parameter for Non-VSAM Data Sets

To automatically create a discrete profile for a new non-VSAM DASD data set (if you don't have the ADSP attribute), specify the PROTECT parameter on the JCL DD statement that identifies the data set (or, for TSO/E, on the ALLOCATE command). Note that the normal reason for a user to use PROTECT instead of ADSP is that most of the user's data sets do not require discrete profiles because they are covered by generic profiles.

Protecting Multivolume Data Sets with Discrete Profiles

To create a discrete profile for a multivolume non-VSAM DASD data set, you must define each volume of the data set to RACF. RACF stores the volume serial numbers in the data set's profile. When the data set is extended to another volume or deleted from a volume, that volume's serial number is automatically added to the profile, or deleted from it.

Alternatively, multivolume data sets can be protected by a generic profile. In this case, the concept of multivolume data sets is irrelevant.

Non-VSAM Data Set Considerations: When a multivolume physical sequential DASD data set is opened for input, RACF does not require that each volume on which the data set resides be defined in the data set profile.

When an existing multivolume physical sequential data set is opened for output, a RACF-protection consistency check is performed. All volumes of the data set that are processed by end-of-volume (when a volume switch occurs) must indicate the same RACF-protection status as the first volume opened. That is, if the first volume is RACF-protected (the DSCB indication is on and the volume is defined in the data set profile), then succeeding volumes are automatically RACF-protected as part of the same volume set; if the first volume is not RACF-protected, then succeeding volumes are not RACF-protected. At end-of-volume, an ABEND occurs if this consistency check fails.

For multivolume non-physical sequential DASD data sets, RACF performs authorization checking for each volume on which the data set resides.

An extension to or deletion from a volume causes RACF to automatically add the volume to or delete it from the data set profile on the RACF data set.

Note: You may not rename a multivolume non-VSAM data set that is RACF-protected with a discrete profile. If the data set is protected with a generic profile, it can be renamed if the new name is also covered by a generic profile.

VSAM Data Set Considerations: For VSAM data sets, extending the data set to a new volume causes RACF to protect the new volume, even though RACF does not add the serial number to the data set profile. The profile for VSAM data sets contains only the volume serial number of the catalog entry for the data set.

For additional information on handling multivolume data sets, see *SPL: RACF.*

Protecting Catalogs

To protect your installation's non-VSAM data sets completely, you must protect the data set catalog in addition to the data sets. RACF provides catalog entry protection for non-VSAM data sets in RACF-protected VSAM catalogs and RACF-protected OS CVOLs.

To protect your installation's VSAM data sets, you **must** RACF-protect the VSAM catalog for those data sets in addition to protecting the data sets.

Figure 4-3 lists the RACF user authorities required to perform catalog operations on a VSAM catalog that is RACF-protected (with UACC=UPDATE).

	VSAM Data Sets			Non-VSAM Data Sets:	
Catalog Operation	RACF-protected	Password-protected	Neither RACF- nor password-protected	RACF-protected	Not RACF-protected
CATALOG	UPDATE to catalog	UPDATE to catalog	UPDATE to catalog	UPDATE to catalog	UPDATE to catalog
RECATALOG	ALTER to catalog or ALTER to data set	ALTER to catalog or PSWD to data set	No authority required	ALTER to catalog or UPDATE o data set	UPDATE to catalog
DELETE/ UNCATALOG/ ALTER	ALTER to catalog or ALTER to data set	ALTER to catalog or PSWD to data set	No authority required	ALTER to catalog or ALTER to data set	No authority required

Figure 4-3. Authorities Required to Perform Catalog Operations on a RACF-Protected VSAM Catalog

For example, using Figure 4-3, assume a user wants to recatalog a RACF-protected non-VSAM data set on a RACF-protected VSAM catalog. The user requires either ALTER authority to the VSAM catalog or UPDATE authority to the data set.

Defining CVOLs to RACF

The naming convention used for OS CVOLs (control volumes) by the RACHECK and RACDEF SVC processing routines is SYSCTLG.Vnnnnnn, where nnnnnn is the volume serial number of the CVOL. That is, when the RACHECK and RACDEF SVCs receive the data set name SYSCTLG as a parameter, they append the qualifier Vnnnnnn to the name.

When you issue the ADDSD command to define an existing CVOL to RACF, use this naming convention and supply the data set name in the form of SYSCTLG.Vnnnnnn. If you do not supply unit and volume serial number information on the ADDSD command and the SET operand is specified or defaulted, the CVOL must be cataloged as SYSCTLG.Vnnnnnn. If you supply unit and volume serial number information on the ADDSD command, ensure that the volume serial number specified on the VOLUME parameter is identical to the volume serial number indicated in Vnnnnnn of the data set name.

Figure 4-4 lists the RACF user authorities required to perform catalog operations on an OS CVOL when it is RACF-protected.

	Non-VSAM Data Sets			
	Non-Generation Data Group		Generation Data Group	
Catalog Operation	RACF-protected	not RACF-protected	RACF-protected	not RACF-protected
CATALOG	No authority required	No authority required	UPDATE to GDG base	No authority required
RECATALOG	UPDATE to data set	No authority required	UPDATE to GDG base	No authority required
UNCATALOG	ALTER to data set	No authority required	ALTER to GDG base	No authority required

Figure 4-4. Authorities Required to Perform Catalog Operations on a RACF-Protected OS CVOL

For example, using Figure 4-4, assume a user wants to uncatalog a non-VSAM data set that is part of a generation data group on a RACF-protected OS CVOL. The user requires ALTER authority to the generation data group base.

Protecting DASD System Data Sets

When you are planning to RACF-protect system data sets, you should consider the following:

- The way in which the system uses the data set
- The way in which your users normally use the data set
- The level of protection you want for the data set

The system data sets can be divided into two categories: data sets for which RACF protection is bypassed when the system accesses those data sets, and data sets for which RACF protection is enforced when the system accesses those data sets.

- **Bypassed RACF Protection**: For a data set of this type, RACF-protection is bypassed when the system accesses the data set to perform its normal system function, but is enforced when a user attempts to access the data set for normal data set operations.

  For example, when the program libraries defined in LNKLSTxx are opened during IPL, RACF protection is bypassed. The system can fetch any program stored in these libraries for the duration of the IPL, but when a user attempts to open one of the libraries, RACF protection is enforced. Assuming SYS1.LINKLIB is defined in LNKLSTxx, it can be RACF-protected giving UPDATE access authority to the system programmers who maintain the data set. You can use UACC=NONE if you do not want anyone other than the system programmers to open the library. The UACC=NONE specification does not prevent any user from executing any program contained in SYS1.LINKLIB, but it does prevent users from, for example, specifying SYS1.LINKLIB as part of JOBLIB or STEPLIB.

  Examples of other system data sets that fall into this category are:

  SYS1.CMDLIB
  SYS1.PROCLIB
  SYS1.LPALIB
  SYS1.SVCLIB
  SYS1.PARMLIB
  SYS1.LOGREC
  SYS1.NUCLEUS
  SYS1.MANn
  SYS1.DUMPnn

  System data sets that are frequently accessed by all users (for example, SYS1.HELP and SYS1.MACLIB) are good candidates for inclusion in a global access checking table.

- **Enforced RACF Protection**: For a data set of this type, RACF protection is enforced when the system accesses the data set for its normal system function on behalf of a specific user. When you protect this type of data set, any user who requests the system function associated with the data set must have a sufficient level of access authority to the data set in order for the function to work correctly.

  For example, if you want to RACF-protect the SYS1.BRODCAST data set, you should give all users UPDATE access authority to the data set because the TSO SEND command opens the SYS1.BRODCAST data set for update. You can give UPDATE access authority by placing 'SYS1.BRODCAST'/UPDATE in the global access checking table. The system programmers who maintain the

data set can be given ALTER access authority via a discrete profile or a fully-qualified generic profile.

Examples of other system data sets that fall into this category are:

Edit utility file
SYS1.HELP
SYS1.MACLIB
SYS1.SAMPLIB

DASDVOL Volume Authority

RACF allows you to define DASD volumes to RACF and authorize users to access the volume table of contents and RACF-protected data sets on those volumes. Thus, a RACF user with DASDVOL authorization to a DASD volume can use the device support facilities to dump and restore data sets on the volume and the SCRATCH function to scratch data sets on the the volume, without requiring authorization to access each of the data sets on the volume.

When using DFDSS (data facility data set services):

- If the DASD volume is defined to RACF in the DASDVOL class and the DASDVOL class is active, RACF-protection and data set password-protection of individual data sets can be bypassed.
- If the DASD volume is not defined to RACF or the system-wide DASD volume protection option is not specified:
  - If a data set on the volume is RACF-protected, access to the data set is authorized through RACF authorization checking or,
  - If a data set on the volume is password-protected but not RACF-protected, access to the data set is authorized by data set password protection.

Scratching DASD Data Sets: A user who has ALTER access authority to a DASD volume can scratch data sets on the volume whether or not the user is authorized access to the data sets. (If the user is not authorized access to the data set, RACF issues message ICH408I even though it allows the data set to be scratched.) When a data set is scratched, the discrete profile for the data set is deleted from the RACF data set or, in the case of a multivolume data set, the volume serial number is removed from the data set profile.

When you use the SCRATCH function, the operation is authorized in one of the following ways:

- If you are authorized by RACF to the data set
- If you are authorized by RACF to the volume
- By data set password protection, if the data set and volume are not defined to RACF, or if the system-wide protection option is not specified

DASD volume authority allows you or your delegate to authorize operations personnel to access only those volumes they must maintain. It also allows you to be selective in placing data sets on DASD volumes to be maintained by operations personnel.

Note that authorization is provided for RACF-protected data sets on authorized DASD volumes only when the system-wide DASD volume authorization option is specified and a data set on the volume is scratched (by the DADSM scratch function).

Moving DASD Volumes Between Systems

When a DASD volume is moved to a system that has RACF installed:

- If the DASD volume is defined on the new system, RACF authorization checking is performed in the normal manner.
- If the DASD volume is not defined on the new system, or the system-wide DASD volume protection option is not specified, RACF protection or password protection is performed for individual data sets.

Protecting Tape Volumes

When the system-wide tape protection option is active, RACF performs authorization checking every time that a tape volume with IBM standard or ANSI labels is accessed (such as with OPEN), whether or not the tape volume is defined to RACF.

A tape volume is RACF-protected when it is defined to RACF for protection. Tape volumes are defined to RACF (1) by an authorized user issuing the RDEFINE command or (2) by the RACDEF macro instruction when the tape option is active and the PROTECT parameter is specified on a JCL DD statement or during EOV processing.

Access to RACF-protected tape volumes is authorized through RACF authorization checking; any tape data set password protection is bypassed. If the tape volume is not RACF-protected or the tape protection option is not active, then access to tape data sets is authorized by password protection.

If a user is authorized by RACF to access a tape volume, then the user has access to all the tape data sets on the volume. Therefore, only place tape data sets that have similar RACF authorization requirements on the same volume.

If you restrict the contents of a tape volume to one tape data set, you can effectively enforce RACF protection at the data set level.

Write-Enable Ring: A message is issued to the system operator to remove the write-enable ring (file protect ring) of a RACF-protected tape volume when the volume is to be processed for input and the user has only READ access authority to the volume.

Tape Volume Protection and Password-Protected Tape Data Sets

When you define a tape volume to RACF with the RDEFINE command, no request is made for the tape volume to be mounted and tape data set password protection is not verified. Therefore, delegate the CLAUTH (TAPEVOL) user attribute only to those administrators responsible for tape volumes.

When a tape volume is defined to RACF via the RACDEF macro instruction, the user or operator must supply the correct password for the password-protected tape data set on the volume to be protected before the tape volume is defined to RACF.

Passwords should be maintained for tape data sets on RACF-protected tape volumes if they contain sensitive information and are to be used (1) on systems that do not have RACF installed or (2) on RACF systems where tape volume protection might not be active.

To maintain passwords for tape data sets, password information can be specified on the LABEL parameter along with the PROTECT parameter on the same JCL DD statement. Also, the password indicators in tape header and trailer labels are set for RACF-protected tape volumes based on password information specified on the LABEL parameter. All password-protected data sets on a RACF-protected tape volume must have the same level of password protection.

Using the PROTECT Parameter for Tape Volume Protection

To automatically define a tape volume to RACF for protection by a discrete profile, specify the PROTECT parameter on the JCL DD statement that identifies a tape data set on the volume. If generic profile checking is active, do not use the PROTECT parameter unless a discrete profile is required for the data set. If the data set is also password-protected, the password must be supplied before the tape volume is RACF-protected.

Multivolume Tape Data Sets

When a multivolume tape data set is opened for input, RACF performs authorization checking for those volumes that are RACF-protected.

When a multivolume tape data set is opened for output, and the first volume is RACF-protected, then all succeeding volumes are RACF-protected as part of the same volume set.

A single profile is maintained in the RACF data set for a tape volume set. Thus all volumes in the volume set share the same access list and the same statistics and auditing options.

GDG Considerations With Tape Volumes

You should use the RDEFINE command to RACF-protect tape volumes on which a generation data group (GDG) resides (or will reside) and then the PERMIT command to authorize users' access to these volumes. Multivolume generations require subsequent RALTER commands with the ADDVOL operand to define the volumes of that generation to RACF as a volume set.

Note that specifying PROTECT on a JCL DD statement that has a generation data set name with a specific volume request results in the volume being defined to RACF the first time that the job is executed and the data set is opened for output. Subsequent uses of JCL with PROTECT result in an ABEND (when the data set is opened) because the tape volume is already defined to RACF.

Tape volumes that contain generations of a GDG, where other generations of the same GDG reside on DASD volumes, should also be defined to RACF individually for protection with the RDEFINE command. Also see "Protecting DASD GDG Data Sets" earlier in this chapter.

Bypassing Authorization Checking for Tape Volumes That Are Not RACF-Protected

When a tape volume is accessed by a resource manager (such as OPEN) and the tape volume option is active, RACF authorization checking (RACHECK SVC processing) performs a search of the RACF data set to determine if the tape volume is defined to RACF.

To reduce the overhead that results from these searches on the RACF data set for tape volumes that are not defined to RACF, consider using RACF options or a RACHECK installation exit routine to differentiate between RACF-protected tape volumes and tape volumes that are not RACF-protected. Four ways to do this are described below.

Using Volume Serial Number Lists: Create a list of the volume serial numbers for RACF-protected tape volumes (and/or non-RACF-protected tape volumes). In your RACHECK preprocessing exit routine, compare the volume serial number of the tape volume to be processed with this list. Then, based on the comparison, bypass or continue with RACHECK authorization checking. This method requires that the list(s) be updated every time a tape volume is defined to RACF (via RDEFINE or RACDEF) or deleted from RACF (via RDELETE or RACDEF).

Using a Volume Serial Number Naming Convention: Establish a specific naming convention for volume serial numbers of RACF-protected tape volumes (and/or non-RACF-protected tape volumes). In your RACHECK preprocessing exit routine, use this naming convention to differentiate between RACF-protected tape volumes and non-RACF-protected tape volumes. Bypass or continue with RACHECK authorization checking based on the naming convention.

Using Global Access Checking: Define entries in the global access checking table to permit access to volumes that all users may access.

Using Generic Profiles: Define generic profiles to control access to tape volumes. Although this does not reduce the search overhead, it simplifies the administrative effort required to protect tape volumes. Do not, however, use generic profiles to protect multivolume tape data sets.

Tape Volume Protection and Bypass Label Processing (BLP)

RACF does not perform authorization checking for tape volumes if (1) the BLP (bypass label processing) option is specified at system generation or JES initialization and (2) BLP is specified on the LABEL parameter of the JCL DD statement for a tape volume. Therefore, the installation should control or limit the jobs that are executed with BLP specified.

If BLP is not specified at system generation or JES initialization, RACF performs authorization checking for a tape volume with IBM standard labels or ANSI labels even if BLP is specified on the LABEL parameter of the DD statement for the tape volume. Users are thereby prevented from specifying BLP for RACF-protected tape volumes in order to bypass authorization checking.

If BLP is specified on the LABEL parameter and BLP is not supported by the system, the tape will be treated as a non-labeled (NL) tape. If a labeled tape is mounted to satisfy this specification, RACF will perform authorization checking and, if the user have sufficient authority, the label will be destroyed.

Tape Volume Protection With Nonstandard Labels (NSL)

Nonstandard label (NSL) processing routines can provide RACF protection for tape volumes with nonstandard labels. You can provide protection by first defining the tape volumes to RACF (with the RDEFINE command or RACDEF macro instruction) and then issuing the RACHECK macro instruction in the NSL processing routine.

Before issuing the RACHECK and RACDEF macro instructions, the NSL routine should ensure that tape volume protection is active. Use the RACHECK macro instruction when a user requests access to a tape volume. Use the RACDEF macro instruction when PROTECT is specified on a JCL DD statement or a multivolume tape data set is extended to another volume (EOV processing). (The JFCBPROT indicator in the JFCB indicates that PROTECT has been specified.)

Tape Volume Protection for Unlabeled (NL) Tapes

If a labeled tape is found when opening the volume for input, the volume is rejected by data management. If a labeled tape is encountered when opening an output tape and the user has ALTER authority to the volume, the volume label is destroyed and RACF protection for the volume is deleted. See *OS/VS Tape Labels* or *MVS/XA Tape Labels* for additional details.

Tape Volume Protection with MVS Tape Label Functions

The ACCLVL parameter on the RACHECK macro instruction specifies tape label access level information for the MVS tape label functions. The access value is passed to the RACHECK installation exit routines. Additional information may be included in an optional parameter.

Protecting Terminals

This topic describes how you can use RACF to control use of the terminals that are connected to your system. RACF can control access to some or all terminals used by TSO, IMS/VS, and CICS/VS users.

To control the use of terminals, you must first activate the RACF terminal class using the SETROPTS command with the CLASSACT operand. The format of the command is as follows:

```
SETROPTS CLASSACT(TERMINAL(READ|NONE))
```

READ is the default if neither READ nor NONE is specified. The specification of READ or NONE establishes a global access authority (for all users) to terminals on your system. If you decide that only selected terminals are to be controlled, you should specify READ. If you decide that **all** terminals are to be controlled (which implies that a terminal can only be used by a user or group that is authorized to it via it's access list), you should specify NONE.

CAUTION: If you specify NONE, be sure that your terminals have been defined to RACF and the appropriate users and groups have been authorized to use them. Otherwise, the terminals will not be accessible.

Once the TERMINAL class is active, users who have CLAUTH authority to the TERMINAL class (you control who these users are) can define terminals to RACF using the RDEFINE command. On systems using VTAM or TCAM, the terminal's node name is the RACF resource name. On systems using BTAM, the resource name for the terminal consists of the the relative line and terminal number. This information can be obtained from the system programmer who is responsible for the Network Communication Program (NCP) system generations for your installation.

With RACF, you also have the capability to select specific RACF groups that can use only those terminals to which the group (or individual users within the group) are specifically authorized. When the group terminal option (NOTERMUACC) is specified (note that TERMUACC is the default) for a group on the ADDGROUP or ALTGROUP command, users of the group can only use those terminals to which they are specifically authorized via the terminal's RACF profile access list.

Note: Global access checking, generic profiles, and list of groups processing are not available with terminal protection. See Chapters 7 and 8 for details on IMS/VS and CICS/VS interfaces with RACF.

Using the TSO LOGON Command With RECONNECT Keyword

TSO provides a line drop facility that enables a user to log onto TSO from another terminal and reconnect to the existing session by issuing the LOGON command with the RECONNECT keyword.

LOGON command processing issues the RACINIT macro instruction during this logon to the new terminal. RACF then performs RACINIT SVC processing for user verification (verifying the user's password and/or operator identification card) and terminal authorization checking (checking the user's authority to access the new terminal). Note that the user cannot change connect groups when the RECONNECT keyword is used.

If verified and authorized, RACF allows the user to resume the interrupted session from the new terminal.

Terminal Information in the Accessor Control Environment Element (ACEE)

After a reconnection, terminal information in the RACF ACEE represents the original terminal and not the new (current) terminal. RACF updates the ACEE with the correct terminal information for the current terminal the next time RACF performs RACDEF or RACHECK SVC processing. This ensures that the RACDEF and RACHECK installation exit routines have correct information in the ACEE for the current terminal.

Note: The installation should be aware that the terminal information in the ACEE may not represent the current terminal. This can occur (1) during the time between a reconnection and the processing of a subsequent RACDEF or RACHECK SVC or (2) if the terminal authorization checking option is turned off between a logon and logon reconnect processing.

Therefore, before installation-written routines (such as commands) access terminal information in the ACEE, they should ensure that the terminal information represents the current terminal by matching the terminal identifier in the ACEE with the current terminal identifier in the TSB (terminal status block).

Protecting General Resources

This topic describes considerations related to the protection of general resources via generic profiles. In addition, information is included on granting resource access authorities.

Protecting General Resources with Generic Profiles

You can use a *generic profile* for most cases in which related resources have similar access-authorization requirements (as well as similar names) and belong to the same class; that is, to one of the general resource classes defined in the class descriptor table. (See "Specifying Generic Profile Checking" in Chapter 5 for more information.)

Rules for Generic Profiles for the General Resource Class: Following are the rules for generic profiles for the general resource class.

- The generic profile name must consist of at least one character. The name can contain any number of characters, each character being either a regular (non-generic) character or a generic % (percent) character, optionally followed by a generic * (asterisk) character.
- A % generic character in the generic profile name matches any non-blank regular character in the resource name.
- The asterisk must be in the last character of the profile name.
- An asterisk standing alone is permitted as, for example, in the profile name "*."
- The characters in the generic profile name, excluding the asterisk, must exactly match the characters in the protected resource's names. The profile "ABC*," for example, protects resources "ABCD" and "ABCXY" but not "ABXD."
- The asterisk represents any string of zero or more regular characters. That is, the profile name "ABCD*" matches both the resources "ABCD" and "ABCDGHKFD."
- Generic profiles can be defined for resource member classes, but not for resource group classes.

Following are some examples of generic profiles for the general resource class:

Generic Profile Name	Resource Names That Match	Resource Names That Do Not Match
ABC%	ABCD ABCX	ABCDE ABC
A%C	ABC ADC	ABBC ADFC
ABC*	ABC ABCD ABCDEF	

Generic Profile Checking of General Resources: The rules for access-authorization checking of generic profiles for general resources are the same as for the DATASET class. See generic profile checking under "Protecting DASD Data Sets."

Granting Resource 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 universal access authority (UACC).

UACC is the default access authority for a 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. These users include those not defined to RACF.

Valid authorities you can specify with UACC or specifically assign to users or groups are listed in Figure 4-5.

Access Level	Description
ALTER	For discrete profiles, the specified user or group has control over the resource and the resource profile, and can authorize other users and/or groups to access the resource. For generic profiles, the specified user or group has control over the resource and can allocate data sets controlled by this generic profile; however, only the profile owner has full control over the resource profile and can authorize other users and/or groups to access the resource.
CONTROL	With this level of access authority, used only for VSAM data sets, the specified user or group has access authority that is equivalent to the VSAM control password. (The VSAM control password allows the user to perform control-interval access and to retrieve, update, insert, or delete records in the VSAM data set it protects.)
UPDATE	The specified user or group is allowed to access the resource for the purpose of reading or writing.
READ	The specified user or group is allowed to access the resource for the purpose of reading only.
NONE	The specified user or group is not permitted to access the resource.

Note: The ALTER and UPDATE levels of access authority apply to DASD data sets, DASDVOL, and TAPEVOL classes; the CONTROL level of access authority applies to DASD data sets only. READ and NONE apply to DASD data sets and IBM-supplied general resource classes. As the security administrator at your installation, you can use these definitions or assign your own meanings for any resource class you define.

Figure 4-5. Resource Access Levels of Authority

Ownership and Access Authorities for General Resources

This section describes ownership and access authorities for general resources, and includes some suggestions for assigning and checking authorization to these resources.

General Resource Ownership

Each general resource defined to RACF requires a RACF-defined user or group to control the profile and access list for the resource. This ownership is assigned when the resource is defined to RACF. As owner, the user can modify, list, and delete the profile, or name another user to become the owner. Note that ownership of a resource profile does not mean that the owner can automatically access the resource. In order to access the resource, the owner must still be authorized to the resource.

General Resource Access Authorities

When you permit other users and groups to access RACF-protected general resources, you give them one of the following access authorities:

ALTER Access Authority - Users and groups with ALTER access authority in discrete general resource profiles have full authority over both the resource and the profile, including the access list. (Generic profiles can only be modified by their owners or by group-SPECIAL users.)

Full authority over tape volumes includes the ability to perform input/output processing for data sets on the tape volume and to create or destroy tape volume labels through OPEN or end-of-volume operations.

Full authority over DASD volumes includes the ability to scratch RACF-protected data sets on the volume.

CONTROL Access Authority - Has meaning only for VSAM data sets. For tape volumes and DASD volumes, CONTROL implies UPDATE. For other IBM-defined resources, CONTROL implies READ.

UPDATE Access Authority - The specified user or group is authorized to access the resource for the purpose of reading or writing.

READ Access Authority - The specified user or group is authorized to access the resource for the purpose of reading only.

NONE Access Authority - The specified user or group is not permitted access to the resource.

Note: These access authority definitions apply only to the IBM-defined general resource classes. The installation can use these definitions or assign its own meanings for any resource class it chooses to define; ALTER authority, however, is not sufficient to modify a generic profile. See "Access Authorities for Generic Profiles" earlier in this chapter.

Universal Access Authority for General Resources

Each general resource that you protect with RACF requires a level of universal access authority. The universal access authority you specify applies to users and groups who are not on the access list for the RACF-protected resource. This includes users not defined to RACF.

If you specifically assign a user or group an access authority to the resource of ALTER, CONTROL, UPDATE, READ, or NONE, this specified authority overrides the UACC specified for the resource.

Suggestions for Assigning Access Authorities

To retain full control over a resource profile, you should give ALTER authority only to the user responsible for the resource.

Tape Volumes - To retain full control over a tape volume, specify a universal access authority of NONE for the volume. Note that, when a tape volume profile is created through the RACDEF SVC, an installation exit routine can specify the UACC value and an access list directly or can name an existing profile to be used as a model.

DASD Volumes - Access authorities to DASD volumes allow RACF users and groups to perform online DFDSS functions and to scratch data sets on the DASD volume. These authorities are also useful for operations personnel who are responsible for maintaining DASD data sets on a volume basis. Users need not be authorized access to any RACF-protected data sets on the volume nor have the OPERATIONS attribute.

Terminals - To restrict users of a group to specific terminals, set the group terminal option (NOTERMUACC) for the group via the ADDGROUP or ALTGROUP command and authorize the group to the selected RACF-defined terminals with READ access authority via the PERMIT command.

Authorization Checking for RACF-Protected Resources

This section describes the checks that RACF makes to authorize users' access to RACF-protected resources. For the following topics, list-of-groups checking for RACF-protected resources can be activated with the SETROPTS command.

Authorizing Access to DASD Data Sets, Tape Volumes, DASD Volumes, and Other Classes That Use the RACHECK Interface

When a user requests access to a RACF-protected DASD data set, tape volume, or DASD volume, the request is allowed based on the identity of the user and whether the user has been permitted sufficient access authority to the resource.

RACF permits access to a RACF-protected DASD data set, tape volume, or DASD volume by making the following checks in the sequence shown.

- If the user is a started task with the privileged attribute, access is granted (unless the CSA option was specified).
- If global access checking is active, the global access table is searched. If a matching entry is found that allows access to the resource, access is granted.
- For DASD data sets, the userid of the requesting user equals the high-level qualifier of the data set name.
- The user's access authority is checked. If the specified access authority is at a lower level than that required to access the resource, access is denied.

- The group to which the requesting user is currently connected is specifically authorized access to the resource. If the GRPLIST option on the SETROPTS command is active, all groups to which the requesting user belongs are compared with the access list. RACF determines the highest access authority and denies access if the selected group's access authority is less than that requested.

- The requesting user has the OPERATIONS (or group-OPERATIONS) attribute and OPERATIONS access is allowed for the class.

- The universal access authority (UACC) for the resource provides sufficient access authority for the requesting user to access the resource.

- SMF log records and/or messages may be generated, depending on the options in effect and whether access was granted or denied.

RACF performs authorization checking when a resource manager (such as OPEN, EOV, or SCRATCH) that controls the protected resource issues the RACHECK macro instruction. If RACHECK processing determines that the requesting user is authorized access to the resource, then RACHECK returns a "successful" return code to the resource manager. The resource manager then allows the request to complete.

If RACHECK processing determines that the requesting user is not authorized to access the resource, then RACHECK returns an "unauthorized" return code to the resource manager. The resource manager then fails the request.

The RACHECK preprocessing and postprocessing exit routines are available during authorization checking.

Authorizing Access to RACF-Protected TSO Terminals

When a RACF user logs on to TSO, RACF performs authorization checking to verify that the user is permitted use of a RACF-protected terminal. RACF performs terminal authorization checking during RACINIT processing at the same time as user identification and verification are performed.

Note: The system-wide terminal option must be in effect for RACF to perform terminal authorization checking.

RACF authorizes use of a RACF-protected terminal by making the following checks in the sequence shown.

- The requesting user is specifically authorized access to the terminal with at least READ access authority. (Use of the terminal is denied if the user's access authority is NONE.)

- The group that the user specifies at logon is specifically authorized access to the terminal with at least READ access authority. (Use of the terminal is denied if the group's access authority is NONE.)

- The universal access authority assigned for the terminal is at least READ access authority and the group specified by the user at logon is authorized to access terminals using the universal access for terminals. (That is, the group terminal option is not set.)

  Note that the universal access for defined terminals is set by the RDEFINE or RALTER command, and for undefined terminals by the SETROPTS command.

The RACINIT preprocessing, RACINIT postprocessing, and "new password" exit routines are available during terminal authorization checking.

Authorizing Access to RACF-Protected IMS/VS and CICS/VS Transactions

You can control access to a given IMS/VS or CICS/VS transaction by defining the transaction to RACF as a resource. (This also applies to any other applications that use the RACLIST/RACHECK interface.) To access this transaction, a user must have the required authority. Authority verification is performed by the FRACHECK routine. The user is authorized to the transaction only if one of the following is true:

- The user is in the access list with at least READ authority
- The user is not in the access list but the user's current connect group is in the access list with at least READ authority
- Neither the user nor the user's current connect group are in the access list but the universal access authority for the transaction is at least READ.

Otherwise, the user is denied access to the transaction. If reverification is required for the transaction, the user must also enter the SIGN ON password with the transaction request.

Note: RACLIST is used to produce an in-storage list of profiles. RACLIST resolves conflicts when more than one profile covers a resource.

When using RACHECK, global access checking is not performed, RACF does not create messages or log SMF records, and changes that are made to profiles do not take effect immediately.

See Chapters 7 and 8 for details on IMS/VS and CICS/VS interfaces with RACF.

Authorizing Access to RACF-Protected Applications

You can control access to a given application (for example, IMS/VS or a user-developed application) by defining that application to RACF as a resource in the APPL class. To use a RACF-defined application, a user must have the required authority. Authority verification is performed by the RACINIT SVC when the application name is passed to RACF on the RACINIT SVC. The user is authorized to the application only if one of the following is true:

- The user is in the access list with at least READ authority

- The user is not in the access list but the user's current connect group is in the access list with at least READ authority
- Neither the user nor the user's current connect group are in the access list but the universal access authority for the application is at least READ.

Otherwise, the user is denied access to the application.

Note: Global access checking, generic profiles, and list of groups processing are not available with application checking.

See Chapters 7 and 8 for details on IMS/VS and CICS/VS interfaces with RACF.

Chapter 5. Selecting RACF Options

This chapter separates the options available with RACF into the following categories:

- Selecting options with SETROPTS
- Selecting options with ICHSECOP
- Using started procedures
- Encrypting RACF user passwords

Selecting Options with SETROPTS

This section describes the following options, which you specify on the SETROPTS command:

- Universal access authority for terminals
- General resource protection
- Generic profile checking
- Global access checking
- Maximum password change interval
- Password syntax rules
- List-of-groups authority checking
- Refreshing of in-storage generic profile and global access checking lists
- Extended password processing
- Data set modeling options
- Bypassing automatic data set protection (ADSP)
- Bypassing RACINIT statistics
- Bypassing resource statistics
- Logging RACF command and RACDEF SVC activity
- Use of real data set names in messages and SMF records
- Bypassing logging of activity of users with the SPECIAL attribute
- Bypassing logging of RACF command violations
- RACF protection for data sets with single-level names
- JES2 or JES3 RACF support

For a complete description of the SETROPTS command, see the *RACF Command Language Reference*.

Universal Access Authority for Terminals

If you have the SPECIAL attribute, you can indicate the universal access authority to be used when users attempt to log onto TSO from terminals that are not defined to RACF. You specify this option with the TERMINAL operand of the SETROPTS command.

The initial default for universal access authority is READ.

General Resource Protection

If you have the SPECIAL attribute, you can specify that access authorization checking is to be provided for classes of general resources specified in the class descriptor table (CDT). You specify this option with the CLASSACT operand of the SETROPTS command.

If you have the SPECIAL attribute, you can also specify the NOCLASSACT operand on the SETROPTS command. This indicates that no access authorization checking is to be performed. However, resource profiles can still be defined to RACF via the RDEFINE command.

The initial default for general resources is no resource protection.

Generic Profile Checking

If you have the SPECIAL attribute, you can enable or disable generic profile checking either on a class-by-class basis or for all classes. You specify this option with the GENERIC and NOGENERIC operands of the SETROPTS command.

If generic profile checking is temporarily disabled (during RACF data set maintenance, for example), but generic command processing is active, all RACF command processors may still work on generic profiles. You control this option with the GENCMD and NOGENCMD operands of the SETROPTS command.

The initial system defaults are no generic profile checking and no generic profile command processing.

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 control this option with the GLOBAL and NOGLOBAL operands of the SETROPTS command.

When you use the SETROPTS command to activate (or reactivate) global access checking for a class, the in-storage global access checking tables are built or updated. 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.

The initial system default is no global access checking.

Maximum Password Change-Interval

The password change-interval specifies the number of days that a user's password is to remain valid.

If you have the SPECIAL attribute, you can specify a maximum password change-interval (within the range of 1 to 254 days). You specify this maximum value in the INTERVAL sub-operand of the PASSWORD operand of the SETROPTS command; it becomes effective immediately as:

- A default value for new users who you define to RACF via the ADDUSER command
- An upper limit for users who specify the INTERVAL operand on the PASSWORD command

The initial system default for the change-interval is 30 days. When users are defined to RACF and have access to the system, the users 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 this option is invoked, the change-interval values in existing user profiles are not modified. RACINIT uses the smaller of the two values.

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 RULEn sub-operand specified by the PASSWORD operand of the SETROPTS command.

List-of-Groups Authority Checking

You can supplement the normal RACF access authority checking by allowing all groups that a userid is a member of to enter in the access list checking process. This process replaces the checking that compares the current connect group with the resource's access list. If more than one group is found in the access list, the highest access authority is used, including authorities conferred by group-SPECIAL, group-AUDITOR, or group-OPERATIONS that a user may have for any groups within this user's list of groups. (See Chapter 3 for details on the limits of authority.)

If you have the SPECIAL attribute, you can specify list-of-groups checking by using the GRPLIST option of the SETROPTS command. To use 'current-connect-group' checking, specify the NOGRPLIST option on the SETROPTS command.

Refreshing In-Storage Generic Profile and Global Access Checking Lists

If you have the SPECIAL attribute, you can initiate the refreshing of the generic in-storage profile list during a job's execution by specifying the GENERIC keyword of the SETROPTS command. The GENERIC keyword causes an indicator in the RACF CVT to be set for the class(es) specified. The refresh is done the next time (after the indicator is set) the generic-profile search routine is invoked for the indicated class(es).

Refreshing of generic profiles used with the FRACHECK service routine is only done when a RACLIST CREATE is issued for the class.

If you specify NOGENERIC on the SETROPTS command, no existing generic profile lists are deleted. The profile lists are not used, and are deleted at the end of the job or when you again specify GENERIC, and rebuilding of the lists takes place as usual.

Note: You must have the SPECIAL attribute to specify the GENERIC option by itself. However, when you issue the REFRESH keyword with the GENERIC keyword (to effect a refresh without altering the status of the system), less authority is required. That is, you must have the group-SPECIAL, group-AUDITOR, group-OPERATIONS, AUDITOR, or OPERATIONS attribute, or you must have CLAUTH authority for the classes specified. The REFRESH keyword also works with the GLOBAL option.

Refresh on Shared Systems: As with list-of-groups and global access checking, the generic profile refresh operation applies only to the system on which you issue the SETROPTS command; in a shared system, you must issue the SETROPTS command on all systems in order to have refresh done on all systems.

Extended Password and Userid Processing

If you have the SPECIAL attribute, you can activate the WARNING/NOWARNING, HISTORY/NOHISTORY, REVOKE/NOREVOKE, INACTIVE/NOINACTIVE, and INTERVAL options.

Use the PASSWORD option on the SETROPTS command to provide the following functions:

- The WARNING subkeyword allows you to specify when a password expiration message should be issued to a userid. If you specify WARNING, RACF issues a message each time the user accesses the system a specified number of days before the password expires. If NOWARNING is in effect, RACF does not issue a warning message before a password expires.

- HISTORY allows you to specify the number of previous passwords to be saved for each user and compared with an intended new password. If there is a match, RACF rejects the intended new password. If NOHISTORY is in effect, RACF does not save previous passwords.

- REVOKE allows you to specify how many consecutive password verification attempts RACF is to permit before it revokes a userid. After RACF revokes the userid, you can activate the userid with the RESUME operand of the ALTUSER command if you have the SPECIAL attribute. If NOREVOKE is in effect, consecutive invalid passwords are ignored.

- The INTERVAL subkeyword controls the maximum interval between password changes.

The INACTIVE keyword of the SETROPTS command causes RACF to revoke the user's right to use the system if the userid has remained unused beyond a specified number of days. If NOINACTIVE is in effect, RACF does not check the userid against an unused userid interval.

If NOINITSTATS is in effect, the INACTIVE, REVOKE, HISTORY, and WARNING options cannot be used.

Data Set Modeling Options

The MODEL operand of the SETROPTS command allows you to automatically supplement the information normally placed in new RACF data set profiles by ADSP, PROTECT, or ADDSD. Modeling can be effective for user, group and GDG data sets on an individual userid or group name basis. You control this processing with the MODEL(USER), MODEL(GROUP), and MODEL(GDG) operands of the SETROPTS command. To specify this option, you must have the SPECIAL attribute.

Bypassing the Automatic Data Set Protection (ADSP) Attribute

If you have the SPECIAL attribute, you can specify that RACF ignore the ADSP attribute when RACF-defined users log on to TSO or submit a batch job for execution. You do this with the NOADSP operand of the SETROPTS command. You can reinstate normal ADSP processing with the ADSP operand.

The initial system default is ADSP processing.

Bypassing RACINIT Statistics Collection

If you have the SPECIAL attribute, you can request that RACF bypass the recording of statistics available during RACINIT processing. By bypassing RACINIT statistics, you can significantly reduce RACF data set I/O activity. You specify this option on the NOINITSTATS operand of the SETROPTS command.

The statistics you can bypass include:

- 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

If you have the SPECIAL attribute, you can also specify the INITSTATS operand on the SETROPTS command, which indicates that RACINIT statistics are to be recorded.

The initial system default is that RACINIT statistics are to be recorded.

Bypassing Resource Statistics Collection

If you have the SPECIAL attribute, you can request that RACF bypass the recording of statistical information for the DATASET class and classes defined in the CDT. You specify this option on 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 has accessed the resource

If you have the SPECIAL attribute, you can also specify the STATISTICS operand on the SETROPTS command, which identifies the classes that are to have statistical information recorded.

The initial default is that statistical information for all resource classes is to be recorded.

Logging of RACF Command and RACDEF SVC Activity

If you have the AUDITOR attribute, you can specify the classes for which all detected accesses to the RACF data set by RACF commands and the RACDEF SVC are to be logged. You specify this option on the AUDIT operand of the SETROPTS command; it becomes effective immediately.

The classes that you can specify in the AUDIT operand and the commands and RACDEF SVC that will be logged for each class are:

User	Group	Dataset	CDT Entries
ADDUSER	ADDGROUP	ADDSD	PERMIT
ALTUSER	ALTGROUP	ALTDSD	RACDEF SVC
CONNECT	CONNECT	DELDSD	RALTER
DELUSER	DELGROUP	PERMIT	RDEFINE
PASSWORD	REMOVE	RACDEF	SVC RDELETE
REMOVE			

If you have the AUDITOR attribute, you can also specify the NOAUDIT operand on the SETROPTS command, which identifies the classes that are not to have RACF command and RACDEF SVC activity logged.

The initial default is that RACF command and RACDEF SVC activity is not to be logged.

Note: If you have the AUDITOR attribute, you can specify on the UAUDIT operand of the ALTUSER command that all RACHECK and RACDEF SVCs issued for the user and all RACF commands (except LISTGRP and LISTUSER) issued by the user are to be logged.

Real Data Set Names in Messages and SMF Records

You use the REALDSN keyword of the SETROPTS command to specify that the actual data set names given on the RACHECK and RACDEF macros are to be put into any SMF log record and operator messages. This will ensure that log printouts and operator messages identify data sets by their real names, rather than by the data set names as modified by installation exit routines (to make them conform to RACF naming conventions). To specify this option, you must have the SPECIAL attribute.

This option has no effect on single-level data set names (both SYSCTLG and others) whose real data set names will continue to be the prefixed ones. This option applies only to name conversions made by the naming conventions table or by installation exit routines.

Bypassing Logging of Activity of Users with the SPECIAL Attribute

If you have the AUDITOR attribute, you can request that RACF bypass logging caused by all RACF commands issued by users with the SPECIAL attribute. You specify this option on the NOSAUDIT operand of the SETROPTS command.

If you have the AUDITOR attribute, you can also specify the SAUDIT operand on the SETROPTS command, which indicates that activity of users with the SPECIAL attribute is to be logged (except LISTDSD, LISTGRP, LISTUSER, RLIST, and SEARCH).

The initial system default is that activity of users with the SPECIAL attribute is to be logged.

Bypassing Logging of RACF Command Violations

If you have the AUDITOR attribute, you can request that RACF bypass logging of all violations detected by RACF commands during RACF command processing. You specify this option on the NOCMDVIOL operand of the SETROPTS command.

A violation can occur because a user is not authorized to modify a particular profile or is not authorized to enter a particular operand on a command.

Whether or not you choose this option, all issuances of the SETROPTS and RVARY commands are logged. whether or not this option is chosen.

Having the AUDITOR attribute, you can also specify the CMDVIOL operand on the SETROPTS command, which indicates that RACF command violations are to be logged (except LISTGRP, LISTUSER and SEARCH).

The initial default is that RACF command violations are to be logged.

RACF Protection for Data Sets with Single Level Names

You can RACF-protect data sets that have names consisting of only a single qualifier (that is, single-level names). Using the PREFIX operand of the SETROPTS command, you can define a qualifier that RACF will automatically prefix to the single-level name. The prefixed qualifier will then serve internally as the high-level qualifier. The prefix should be an existing userid or group name, but must not be the same as the high-level qualifier of a multi-level data set name. To specify the PREFIX or NOPREFIX operands, you must have the SPECIAL attribute.

JES2 or JES3 RACF Support

The availability of the JES support described in this section is dependent on the level of the JES programming support installed at your location. To determine the level, see the product specifications for the Job Entry Subsystem (JES) installed on your system(s). When the JES support described herein is not installed, the RACF SETROPTS command will function as described, but the JES function it is designed to control is not available.

JES User Identification Propagation

To reduce the administrative overhead of maintaining RACF userids, passwords, and (sometimes) groupids in the job statements for all batch jobs, JES propagates the current RACF userid and groupid from each already validated RACF user who is submitting a batch job to JES via the JES internal reader interface routine. Jobs submitted in this manner (for example, jobs submitted with the TSO SUBMIT command) are marked as already password-validated so that when the jobs are later processed by the initiator, the propagated userid and groupid are used by RACF to create the ACEE, but password validation is not required.

This means that jobs submitted by a RACF/TSO user are automatically identified with that user and the group the user was connected to when the user submitted the job. Note that neither a password nor any other RACF information is required in the JOB statement. Of course, if a TSO user submits a job for another user, the TSO user must specify that user's userid and password. Also, if a TSO user submits a job that is to be associated with a group other than the present log-on group, the user must specify group information on the JOB statement.

JES Userid Early Verification

For a job that cannot have the validated userid and groupid propagated by JES (for example, a job read in by card reader), the installation can install an exit routine at a point that gets control soon after the job is read. An installation exit routine installed at this point would have the capability to verify that userid data is present on the job statement. The SETROPTS JES(EARLYVERIFY/NOEARLYVERIFY) command determines whether this exit routine gets control.

The EARLYVERIFY specification turns on an indicator in the RACF CVT that JES can test. If the indicator is on and JES is reading a batch job that does not qualify for user identification propagation, a call is made to the system authorization facility (SAF) via the RACROUTE macro.

The RACROUTE macro is issued with the REQUEST=VERIFY and ENVIR=VERIFY parameters. This invocation of SAF does not result in a call to RACF. It is intended to provide, via the SAF MVS router exit, access to optional, installation-developed/installed userid verification logic.

SAF and the MVS router exit are documented in *SPL: RACF*, which also contains more information about using the SAF MVS router exit for early verification processing.

Forcing Batch Users to Identify Themselves to RACF

To prevent unauthorized users from running batch jobs, the installation can optionally specify that all batch jobs require RACF identification. The following RACF command options control this function:

```
SETROPTS JES(BATCHALLRACF|NOBATCHALLRACF)
SETROPTS JES(XBMALLRACF|NOXBMALLRACF)
```

The BATCHALLRACF specification turns on an indicator in the RACF CVT that JES can test. If the indicator is on, JES will fail any batch job that is not to be run under an execution batch monitor (XBM) and does not have propagated user identification or a valid userid and password on the JOB statement. The NOBATCHALLRACF specification turns off this indicator. NOBATCHALLRACF is the default when RACF is installed.

The XBMALLRACF specification turns on an indicator in the RACF CVT that JES can test. If the indicator is on, JES will fail any batch job that is to be run under an execution batch monitor and does not have propagated user identification or a valid userid and password on the JOB statement. The NOXBMALLRACF specification turns off this indicator. NOXBMALLRACF is the default when RACF is installed.

Selecting Options with ICHSECOP

This section describes the following options, specified in the ICHSECOP module:

- Bypassing RACF initialization during IPL
- Selecting the number of resident index blocks
- Disallowing duplicate data set names

To use these options, you must replace the ICHSECOP module supplied by IBM. The version of ICHSECOP you receive from IBM is set so that during IPL, RACF initialization processing is performed (RACF is activated), ten resident index blocks are available, and duplicate data set names are allowed.

The module is used only during IPL. When the module is changed, the changes are not effective until after the next IPL. See *SPL: RACF* for additional details.

Bypassing RACF Initialization Processing During IPL

You can use this option as part of a procedure for bypassing RACF functions at any time after the installation of RACF is complete. When this option is in effect, RACF is inactive for the life of an IPL and the RVARY command cannot be used to make RACF active.

When you use this option, no RACINIT SVC is issued during TSO logon, IMS/VS or CICS/VS sign on, or job initiation processing. If a JOB statement contains the USER, GROUP, and PASSWORD parameters, the system ignores them. TSO reverts to UADS user identification and verification. Also, no one can issue RACF commands.

If a user accesses a RACF-protected resource, the RACHECK SVC is still issued. If you are using any RACF-protected resources on your system, (1) use the SETROPTS command to turn off resource protection prior to IPLing with this option in effect, and (2) to instruct the operations staff about the RACF failsoft messages and intervention requests. See *SPL: RACF* for additional details.

Selecting the Number of Resident Index Blocks

You can select the number of RACF data set index blocks to be made resident in the common service area (CSA). Resident index blocks reduce the amount of I/O that is required to service the RACF data set. Note that this option can also be controlled using the RACF data set name table. See *SPL: RACF* for additional details.

Disallowing Duplicate Data Set Names

You have the option of not allowing duplicate DASD data set names to be defined to RACF. When you use this option, the RACF manager will fail ADDSD command and RACDEF DEFINE macro instruction requests if the data set name is already located in the RACF data set. See *SPL: RACF* for information on how to select this option.

Using Started Procedures

You can associate the names of started procedures with RACF userids and group names. This option is part of a process that allows started procedures, such as JES, to have specific authorization to access RACF-protected resources, such as spool data sets.

Started procedures have system-generated JOB statements that do not contain the USER, GROUP, or PASSWORD parameter. Because only users and groups can be specifically authorized to access RACF-protected resources, this option allows started procedures the same ability.

As with any other userid and group name, those assigned to a started procedure must be defined to RACF using the ADDUSER and ADDGROUP commands.

You may also need to authorize the users and/or groups to the required resources using the PERMIT command. These steps must be done before installing the started procedures module on the system. See the *RACF Command Language Reference* for descriptions of the commands.

Optionally, you can update the started procedures table to contain a generic entry, indicated by an asterisk (*) in the procedure name field. This entry must be the last entry in the table; otherwise it will be ignored. The corresponding userid in this entry can be a valid userid or an equal sign (=). The group name specified in the table entry can be blanks, a valid group name, or an equal sign (=). Note that the userid and group name cannot both contain an equal sign (=) in the "*" procedure name entry of the table because it is not possible to have a RACF user and group that are named identically.

When the started procedures table is searched for a procedure name match, the asterisk (*) entry is used as a match for the procedure name if the asterisk (*) is the last entry in the table and the procedure name was not specifically matched by any other entry in the table. If a userid is specified in the user field of the asterisk (*) entry, that userid is associated with the started procedure name. If an equal sign (=) is specified in the user field, the procedure name is used as userid.

If no group name is indicated (that is, the group name is blank), the default group in the corresponding user profile is used. If a group name is given, it is used. If an equal sign (=) is specified in the group name field, the procedure name is used as group name.

As an additional enhancement, the installation can mark individual entries in the started procedures table to indicate that the procedure be considered "privileged," meaning that all RACHECK calls done for it will be considered successful, without actually performing any checking. No installation exits will be called, no SMF records will be generated, and statistics will not be updated.

When the started procedure is executed, the default group in the user profile (specified on the ADDUSER command) is used if you do not supply a group name in the module. No user verification (password checking) takes place for a started procedure's userid.

The started procedure name is always available to the installation-written exit routines, whether or not the name is coded in the module. It is available in the parameter list for the RACINIT SVC and in the ACEE control block for the RACHECK and RACDEF SVCs.

If a started procedure is executed without associating its name with a RACF-defined userid and group name, the RACF default userid (*) and group name (*) is used for authorization checking. The procedure can access RACF-protected resources if the universal access authority for the resource is sufficient to allow the requested operation.

For information on how to code the started procedures replaceable module, see *SPL: RACF.*

Encryption of RACF User Passwords

By default, RACF passwords and operator identification card (OIDCARD) data are stored in the RACF data set 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 and OIDCARD data. 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 encryption functions:

1. Password/OIDCARD data encryption
2. Password/OIDCARD data comparison

Encryption 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" encryption. That is, data encrypted by RACF can only be decoded if the data is already known. See *SPL: RACF* for additional details.

Comparison means that, given a password (or OIDCARD data) as entered by a user (in clear text form) and given a password (or OIDCARD data) as stored in the RACF data set in encrypted 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 encrypts 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 encrypting the user-entered data against the DES algorithm and against the current masking algorithm, RACF allows the use of existing masked passwords and OIDCARD data until they can be replaced by the DES forms. (Note that this will eventually happen, because whenever a new password is given, it is encrypted via the DES algorithm and stored in the RACF data set in this way. The ICHDEX01 exit routine can be used to change this processing.)

Note that when the user is using TSO, the password is still stored in clear text form in the terminal status block (TSB) in read-protected storage for the life of the session.

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 LPALIB, followed by an IPL.

This should be done on all systems sharing the RACF data set after they have all been converted to a version of RACF that supports the DES algorithm. Once all masked data has been converted to DES-encrypted 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.

Chapter 6. Operating Considerations

This chapter contains some of the operating considerations with which you, the security administrator, should be familiar.

Achieving System Security After the First IPL with RACF Installed

After you have IPLed your system with RACF active for the first time and have included a basic set of profiles in the RACF data set, you can quickly achieve system security. There is one user profile with the userid of IBMUSER and three group profiles with the group names of SYS1, VSAMDSET, and SYSCTLG (where VSAMDSET and SYSCTLG are subgroups of SYS1). IBMUSER is connected to each of the three groups.

IBMUSER is the first userid that you, as security administrator, will use. This user has full authority to use any of the RACF functions or to access any RACF-protected resources through the SPECIAL and OPERATIONS attributes. (Note that IBMUSER does not have the AUDITOR attribute, but can give himself or others this attribute.)

The first thing IBMUSER does when entering the system is change the initial password SYS1 to a new password. This new password prevents any other user from entering the system as IBMUSER.

If you want to be identified by a userid different from IBMUSER, you can use the ADDUSER command at this time to define another user profile. The new profile should have the default group SYS1 and the SPECIAL attribute as well as whatever user attributes and default universal access value you will need to RACF-protect the critical resources of the system. You should also connect the new user profile to the VSAMDSET and SYSCTLG groups with the RACF CONNECT command. Then you should log on or submit a batch job to the system using the new userid (and a new password) and issue the ALTUSER command to set the REVOKE attribute in the profile for IBMUSER to prevent further use of this userid.

To retain the userid IBMUSER, you should use the ALTUSER command to change the default universal access value and user attributes in the IBMUSER profile, if required, to the access value and user attributes that will be needed to RACF-protect the critical resources of the system. You must then re-enter the system in order to make these changes effective.

You can then use the ADDSD command to protect the DASD data sets that are critical to the security of the system.

Using the group SYS1, you can protect such data sets as the RACF data set, SYS1.LPALIB, SYS1.NUCLEUS, and data sets in the LNKLSTxx member of SYS1.PARMLIB. You can use the group VSAMDSET to protect VSAM data sets with names that are generated by the system. VSAM generates data set names using VSAMDSET as the high-level qualifier. You can use the group SYSCTLG to protect OS CVOLs.

After protecting the critical resources of the system, you can continue to carry out the implementation plan for defining users, groups, and resources to RACF and permitting users and groups to access resources. If always call is installed, you can protect the bulk of the installation's resources with generic profiles (1) to minimize the administrative effort involved in defining resources to RACF, and (2), to simplify RACF processing during access authorization checking.

Note: Appendix A provides a sample RACF command session for defining users, groups, and resources.

As soon as users are defined to RACF, they must change the temporary passwords supplied by their RACF administrator to new passwords to secure the system. They can also change the default prefixes in their TSO profiles (using the TSO PROFILE command) to appropriate userids or group names. The users can then add the RACF operands to their JCL JOB statements as needed to access RACF-protected resources.

If you are planning to associate started procedure names with userids and group names, you must be sure the userids and group names are defined to RACF and authorized to access any RACF-protected resources the started procedures require before you replace the started procedures module.

Checking System Security

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 will help you to (1) check the initial steps you took to establish system security, and (2) make additional security checks periodically. The reports that might be particularly useful to you are:

- Selected user attribute reports
- Program properties table report
- Authorized caller table report
- Selected data sets report
- RACF exits report

A short description of each report follows. For more information on these reports and the data security monitor, see the *Auditor's Guide.*

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.

Program properties table report: This report lists all the programs in the program properties table (PPT). The report also indicates whether each program is authorized to bypass password protection and whether it runs in a system key.*:* You can use the program properties table report to verify that only those programs that the installation has authorized to bypass password protection are, in fact, able to do so. (If a program has bypass password protection, RACF does not perform authorization checking for RACF-protected DASD data sets and tape volumes during system operations such as OPEN.) Such programs will normally be communication and data base control programs, and other system control programs. You can also verify that only those programs that the installation has authorized are able to run in a system key.

Authorized caller table report: This report lists the names of all the programs in the RACF authorized-caller table; that is, programs that are authorized to issue RACF supervisor calls (SVCs). The report also indicates whether each program is authorized to issue the RACINIT macro (which performs user verification) and the RACLIST macro (which loads profiles into main storage).

You can use the authorized caller table report to verify that only those programs that are supposed to be authorized to modify an ACEE (access control environment element) are able to issue a RACINIT SVC. This verification is a particularly important security requirement because the ACEE contains a description of the current user. This description includes the userid, the current connect group, the user attributes, and the group authorities. A program that is authorized to issue the RACINIT macro could alter the ACEE to simulate any user.

You can also use this report to verify that only those programs that are supposed to be authorized to access a resident profile on the RACF data set are able to issue the RACLIST macro. Because profiles contain complete descriptions of the characteristics associated with RACF-defined entities, you must carefully control access to the profiles.

Selected data sets report: This report lists the names of selected system data sets and, for each data set, specifies the criterion for selection, the serial number of the volume on which each it resides, whether the data set is RACF-indicated and/or RACF-protected, and the universal access authority (UACC). If a data set meets more than one selection criterion, there is a separate entry in the report for each criterion.

You can use the selected data sets report to determine which system and RACF data sets are protected by RACF and which are not. You can also check whether the UACC associated with each of the data sets is compatible with your installation's resource access control requirements.

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.

RACF Generalization

The goal of RACF generalization is to allow both IBM-supplied and installation-written programs and applications to use RACF as a security mechanism. RACF functions that contribute to generalization are:

- General resource class support
- New class definition
- Resource group support
- Support for multiple users in each address space
- Application authorization
- Resident resource profiles (RACLIST)
- FRACHECK authorization checking routine

These items are described in the following topics.

General Resource Classes

With the exception of the DATASET, USER, and GROUP classes, all resource classes are represented in the class descriptor table (CDT). Each class descriptor contains control information RACF needs to verify that a valid class name has been specified. The IBM-supplied CDT contains class descriptors for the following classes:

- DASDVOL (DASD volumes)
- TAPEVOL (tape volumes)
- TERMINAL (terminals)
- TIMS (IMS/VS transactions)
- GIMS (IMS/VS transaction groups)
- APPL (applications)
- TCICSTRN (CICS/VS transactions)
- GCICSTRN (CICS/VS transaction groups)
- PCICSPSB (CICS/VS program specification blocks or PSBs)
- QCICSPSB (CICS/VS PSB groups)
- AIMS (IMS/VS application group names)
- GLOBAL (for global access checking)
- GMBR (for global access checking)
- DSNR (DB2)

RACF commands and SVCs reference the CDT whenever a class name other than DATASET, USER, or GROUP is received as input.

Note: The class descriptor table must be the same for each system sharing the RACF data set. If they are not the same, you might get unpredictable results when using the SETROPTS command to activate or deactivate RACF classes.

New Class Descriptor Definitions

You can add new class descriptors or modify or delete old class descriptors in the CDT as needed. You accomplish this by one of the following methods:

- Invoke the class descriptor macro ICHERCDE for each resource class and assemble. Linkedit the result to produce the load module used by RACF.
- Linkedit the object module(s) for the class(es) being added or modified together with the existing load module ICHRRCDE in SYS1.LPALIB to produce a new load module.

See *SPL: RACF* for details on how to add or delete class descriptors.

Resource Groups

You can use resource groups to protect, with one RACF profile, a set of general resources (but not data sets) that have the same security requirements. (For example, a given user or group has the same authority to all the resources in the group). Figure 6-1 provides a sample of the creation of resource groups and resource group classes.

To make use of resource groups for a given class of resources (for example, TIMS), you define a resource group class (in this example, GIMS) associated with the given class of resources. To group a set of resources in the given class, you define an entity in the associated resource group class. Each resource in the set is made a member of the grouping entity. When a user is allowed to access the grouping entity (via the PERMIT command), the user's authority is propagated to each member resource. Also, the universal access of the grouping entity is propagated to each resource.

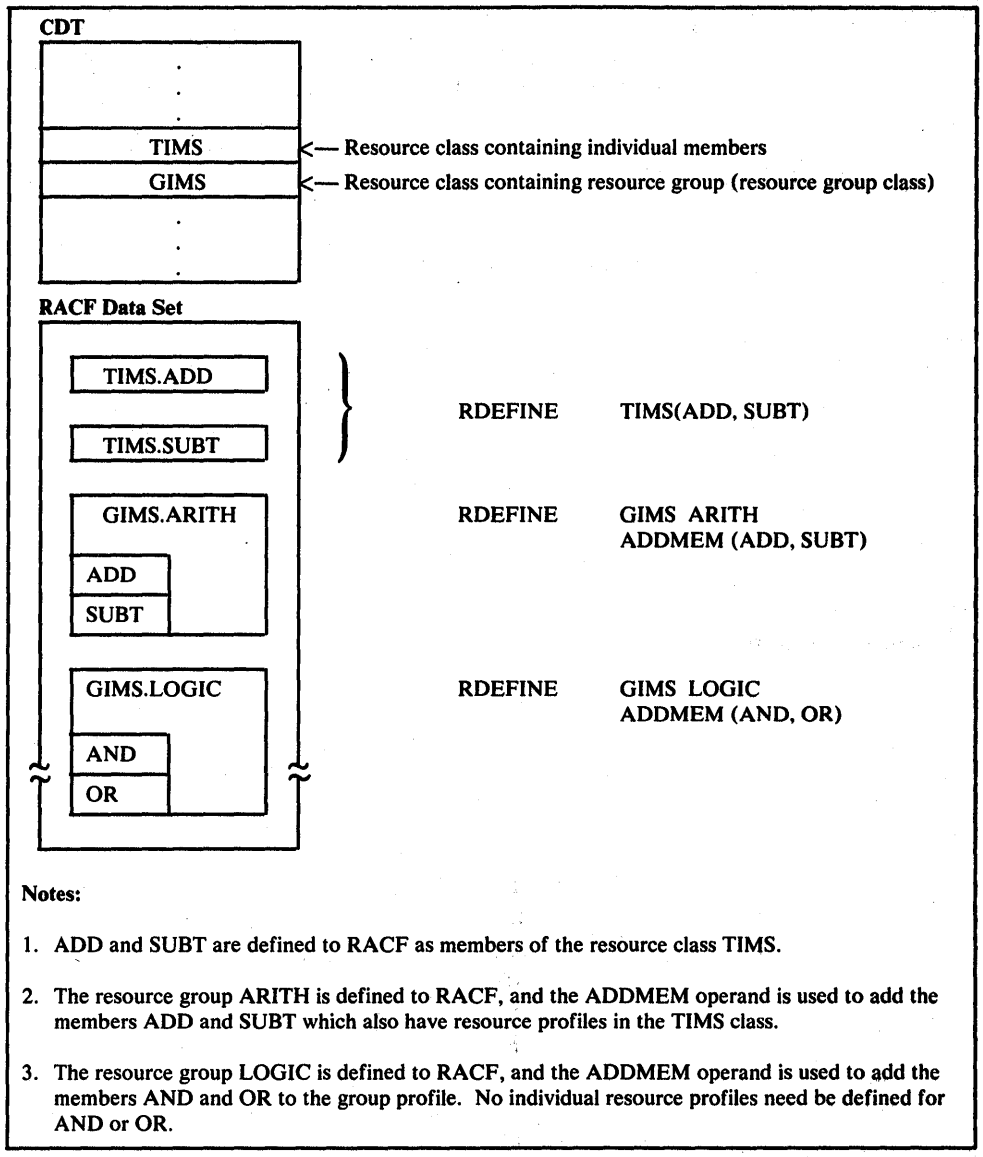


Figure 6-1. Creating Resource Groups

When planning for the use of resource groups, consider the following:

- Resource groups are effective only when used in combination with RACLIST. RACF does not automatically propagate the access list information and universal access of a grouping entity to its member resources when the resources are made members of the grouping entity or when users or groups are allowed access to the grouping entity via the PERMIT command. Instead, this propagation takes place when RACLIST is used to construct in-storage profiles for a given resource class. Resource grouping can therefore be used only for those classes of resources for which the resource manager (for example, IMS) invokes RACLIST prior to performing authorization checking.

- A grouping entity is, in its own right, a RACF-protected resource. Thus, RACF controls access to the entity. Any user accessing that entity must have sufficient authority to perform the desired operation. A resource grouping class is associated with one and only one resource class and cannot be used to group resources from two different classes. Nor can a single resource class be grouped by two different resource group classes.

- You cannot define generic profiles in the resource group classes, but you can include generic names as members of a grouping entity.

Application Authorization

The generalized function of RACF can be used not only to authorize users to an application's resources, but also to authorize users to the application itself, as described earlier under "Authorizing Access to RACF-Protected Applications" in Chapter 4.

Resident Resource Profiles

Applications use the RACLIST SVC to build in-storage profiles for a given class of resources. See *SPL: RACF* for details on RACLIST processing.

FRACHECK Authorization Checking Routine

FRACHECK uses the resident profiles constructed by RACLIST to perform authorization checking. Because FRACHECK performs no logging, gathers no statistics, and issues no SVCs, there is a loss of function compared to RACHECK. Therefore, using FRACHECK is recommended only for applications that have stringent performance requirements. Associated with FRACHECK are two installation exits that can be used to make additional security checks or to instruct FRACHECK to either accept or fail the request.

An installation could write an application that uses RACLIST and FRACHECK for authorization checking on a resource class and associated resource group that the installation defined.

Global Access Checking

Global access checking can be used to allow access, via a dynamically maintained in-memory table, to protected resources that are accessed on a regular basis by many users at an installation. Global access checking cannot fail a request; it can only allow access. If global access checking does not find that the user has the necessary access authority, or if global access checking is disabled, normal RACF processing is done.

The global access checking facility processes access attempts by comparing the access level requested by the user (READ, UPDATE, CONTROL, or ALTER) to the access level associated with the resource in the global access checking table. If the access level of the user's request does not exceed the level specified in the table entry for the resource, global access checking bypasses normal RACF processing. Otherwise, normal RACF processing via discrete and generic profiles is performed.

The names of entries in the global access checking table must conform to the following rules:

- Regular characters in the table must match exactly characters in the resource name, character by character.
- A percent character (%) in the global table entry matches any non-blank, regular character in that position in the resource name.
- For the general resource classes, an asterisk occurring in the last position in the name in the global access table is regarded as a match to any characters in the same or subsequent positions in the resource name.
- For the DATASET class, asterisks can represent one or more qualifiers; as for generic data set profile names, an asterisk in the last position represents any number of qualifiers at the end of the data set name. Unlike a generic data set profile name, an entry in the global access checking table can contain an asterisk representing the high-level qualifier.
- The words &RACUID (RACF userid) and &RACGPID (RACF groupid) are special words that can be used as name qualifiers. For example, the specification

  ```
  &RACUID.*/ALTER
  ```

  matches all data sets that begin with the userid of the user who is accessing the data set. The word &RACGPID allows the current connect group to be used in the same way.

Note that generic command processing must be active in order to add generic names to the global access checking table. Once added, they are used in global access checking whether or not generic access checking is active. You can activate generic command processing with the SETROPTS command.

When using global access checking, consider the following:

- The global access checking table specifies access levels for resources but does not include access lists. Each user's request is compared with the specified access levels.
- The group resource classes are ineligible for global access checking.
- When global access checking allows a request, any OS password processing and prompting that would otherwise have occurred is bypassed.
- When global access checking allows a request, RACF does not perform any logging and does not maintain any statistics.
- RACF bypasses global access checking if a profile is being passed to RACHECK or if the CSA option is used.
- Updated global access checking table entries become effective with the next IPL or after execution of the SETROPTS command with the GLOBAL (classname) operand, with or without the REFRESH option.

RACF creates, maintains, and lists global access checking table entries by using the same commands it uses for general resource profiles: RDEFINE, RALTER, RDELETE, and RLIST. When used for global access checking, the first operand must be the keyword GLOBAL, and the second operand must be the class-name. The ADDMEM and DELMEM operands define the table entries. For example,

```
RDEFINE GLOBAL DATASET ADDMEM(&RACUID.SALES.*/ALTER)
```

defines an entry in the global access checking table specifying that ALTER authority is allowed for any data set whose high-level qualifier is the current user id and whose second-level qualifier is SALES. The access level (READ, UPDATE, CONTROL, ALTER, or NONE) follows each member name separated from the name by a slash ("/"). If no access level is specified, READ is assumed.

As described under "Selecting RACF Options" in Chapter 5, the SETROPTS command activates global access checking. To update the global access checking table (after a change has been made to it), issue the SETROPTS command again.

Refreshing Generic In-Storage Profile Lists

You can use the SETROPTS command to refresh in-storage generic profile lists during execution of a job. See Chapter 5 for information on how to use the SETROPTS command to refresh in-storage generic profile lists.

Restarting Jobs

When a job automatically restarts and returns to a previous checkpoint, RACF repeats user verification and access authorization checking. If the job changed the password on the JOB statement, RACF uses the new password for user verification. But if, meanwhile, the PASSWORD command or another job changes the password, RACF detects an invalid password and fails the job.

When you submit a job for a deferred restart, you should specify your current password on the JOB statement.

For tape volumes, if a user is restarting a job via a deferred step restart that has the PROTECT parameter specified on a JCL DD statement, the job will fail if the tape volume had been defined to RACF before the restart. In this case, remove the PROTECT parameter from the DD statement or delete the tape volume profile (with the RDELETE command) before restarting the job. An automatic step restart (with PROTECT specified) will only be successful if the step abnormally terminated before the tape data set was opened for output and before the tape volume was defined to RACF.

For either an automatic or deferred restart, the user's current access authority (the access authority at the time of the restart) is checked.

RACF Processing and Bypass Password Protection

Programs that execute with bypass password protection (bit 6 is set in the program properties table IEFSDPPT) have the following effect on RACF processing:

- RACHECK macro instructions are bypassed by OPEN, EOV, checkpoint/restart, DADSM rename, and DADSM scratch. Thus, RACF does not perform authorization checking for RACF-protected DASD data sets and tape volumes.
- RACDEF macro instructions are bypassed by OPEN, EOV, and DADSM scratch. This can cause the following conditions:
  - After scratching a RACF-protected DASD data set or performing tape label destruction (by OPEN or EOV) for a RACF-protected tape volume, the RACF profile is not deleted from the RACF data set.
  - A DASD data set defined to RACF by DADSM allocate (as a result of ADSP or PROTECT) that extends to subsequent volumes will not have those volumes defined in the data set's discrete profile.

Controlling Access to RACF Passwords

Installation personnel should ensure that the security of RACF user passwords is not violated.

You should restrict the operator's use of the JES3 operator commands. Using JES3 commands, the system operator can display JES3 data areas that contain both the current and new RACF passwords associated with a job, even though these passwords are in a masked format. (When a user submits a job and supplies RACF passwords on the JOB statement, JES3 stores them for the life of the job.) You can monitor the operator by logging the operator commands or by having the operator submit all console listings for review by security personnel.

It is also possible for the operator to display password information when displaying real storage at the console. Again, the installation should monitor the operator's activities to ensure passwords remain secure.

In addition, you should restrict access to SVC dumps and stand-alone dumps, which might contain password information.

You should restrict the TSO SUBMIT command capability so that users can only submit jobs that execute under their own userids. In this way, a user doesn't need to know anyone else's password. You can enforced this by using an installation exit routine in the SUBMIT command processing.

If JES support for user identification propagation is installed, batch jobs being submitted by TSO users need not have any RACF identification information

(userid, groupid, and password) in their JOB statements. However, if JES support is not installed, the following considerations should be made:

TSO Extensions (5665-285) must be installed if you wish to use SUBMIT command parameters for batch jobs that access RACF-protected resources.

If a user specifies //DD DATA and neglects to delimit the data (with /* or DLM specification) when submitting a batch job via a card reader or RJE work station, subsequent jobs will be read as part of the user's data until a delimiter is read. You should be aware that if this situation occurs, RACF userids, group names, passwords, and resource names from the following job's JCL become available to the user who failed to supply a delimiter. The installation should use SMF or JES installation-written exit routines to restrict the use of the //DD DATA statement to reduce this security exposure.

Authorizing Only RACF-Defined Users Access to RACF-Protected Resources

If the universal access authority (UACC) for a RACF-protected resource is other than NONE, then:

- Non-RACF-defined users can access the RACF-protected resource with the specified level of universal access.
- RACF-defined users (who have NONE access authority to the RACF-protected resource) can access the resource by submitting a batch job without specifying the USER parameter on the JCL JOB statement.

These accesses to RACF-protected resources can be prevented by the RACHECK preprocessing exit routine that fails RACHECK processing for users who have entered the system using the RACF default userid.

Note that this RACHECK processing requires RACF-defined users to identify themselves (via the USER parameter) on batch jobs that access RACF-protected resources and prevents non-RACF users from accessing RACF-protected resources.

JES2 Execution Batch Monitor

If XBMALLRACF is set, jobs running under an execution batch monitor require user information, either on the JOB statement or propagated from the submitter. However, when jobs running under an execution batch monitor are executed, they do not have individual ACEEs that reflect the user data on the JOB statements. Instead, these jobs are identified with the ACEE of the monitor. That is, the jobs running under the execution batch monitor can use only RACF-protected resources that are accessible to the monitor. Note that the monitor, itself, can be run as a job or as a started procedure and identified to RACF and authorized to protected resources in the usual manner. See "JES2 or JES3 RACF Support" in Chapter 5 for details on the XBMALLRACF specification.

Using TSO When RACF is Deactivated

While RACF is deactivated (as a result of issuing the RVARY command), users can still log on to TSO, but RACF does not do any processing. (For example, a user who has the ADSP attribute can allocate DASD data sets, but RACF will not be called to define discrete profiles for those data sets.)

If the TSO user has the WTPMSG option active, the user will receive messages in the session indicating the identity of the resource being defined or accessed.

Therefore, you should let users who have the ADSP attribute know when RACF is not active so that they are aware that profiles for newly-created data sets are not being defined to RACF.

When RACF is reactivated, you should advise RACF users to log off TSO and log on again to ensure they are connected to their proper RACF group.

Using the TSO EDIT Command

If a user edits a RACF-protected data set to which the user has only READ access authority, a failure occurs when the user attempts to save the data set. To issue the SAVE command, the user must have at least UPDATE access authority to the data set.

Service by the IBM PSR and CE

When an IBM programming support representative (PSR) and customer engineer (CE) require access to the system for servicing, they must be defined to RACF if they need to access RACF-protected data sets for servicing. Also, they need the appropriate access authority to these data sets.

You can define user profiles for the PSR and CE 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 the PSR or CE to enter the system. (The REVOKE and RESUME operands of the ALTUSER or CONNECT command alter the REVOKE attribute.)

Removing RACF From Your System

You can use the following procedure to remove RACF. This procedure assumes that RACF is active (not deactivated by RVARY command). The RACF commands used in the procedure must be issued by a user with the SPECIAL attribute. The user may need to have the TSO MOUNT attribute.

Before removing RACF, you must prevent users from adding RACF-protection to DASD data sets while you are deleting data set profiles from the RACF data set. To do this, you can revoke all users by using the ALTUSER command.

1. Do the following to remove RACF-protection from DASD data sets:

   - Issue the SEARCH command (with the NOMASK operand) to locate all RACF-protected data sets. Request that a CLIST be created containing a DELDSD command for each data set name located in the scan.

     Note: Duplicate data set names require special handling because they require the VOLUME operand, but the SEARCH command does not put volume information in the CLIST.

   - Use the TSO EXEC command to execute the DELDSD commands in the CLIST data set. The DELDSD commands will remove each data set profile from the RACF data set and turn off the RACF indicator in the DSCB for a non-VSAM data set or in the catalog for a VSAM data set. This step might involve mounting several volumes to complete the job.

   Deactivate the RACF function in your system using the control information and instructions in the RACF program directory that you received from IBM. In general, the procedure consists of the following:

   - Use the procedure in the RACF program directory to update the libraries containing the RACF load modules.
   - Re-IPL to initialize the system without RACF.
   - Alter the MSTRJCL if necessary.
   - Uncatalog and scratch the RACF data set, if desired.

2. Remove the RACF operands from JCL, JOB, and DD statements.

Failsoft Processing

The operator has limited involvement with RACF failsoft processing. Resource access checking requests are processed using global access checking tables, RACLIST in-storage profiles, or a supplied profile, if any of these are present. (Generic profile checking is not done because an integrity exposure arises if a generic profile is used to allow access to a resource when, in fact, there exists a discrete profile that would have disallowed access, had that profile been retrieved.)

RACHECK and RACDEF preprocessing installation exits are called during failsoft processing. (The postprocessing exits are not called.) This action frees the installation to define its own version of failsoft processing. A resource access can be allowed, denied, or made to continue normal failsoft processing.

Logging is done as when RACF is active, for all resource accesses that are automatically allowed or denied by failsoft processing. Logging is also done for all accesses that are allowed or denied by the operator.

If no global access checking tables are present, no RACLIST in-storage profiles are present, and no profile has been supplied, the preprocessing installation exits will be called first, and then failsoft processing continues as follows:

- RACHECK:

  1. For started tasks, RACF issues an information message to the operator to describe the name and access mode of the resource. RACF permits the access attempt.

  2. For TSO sessions, RACF issues the information message and, if the high-level qualifier of the data set name does not match the TSO userid, RACF also issues an operator intervention message to request permission to allow access to the resource. If the system operator gives a negative response to a request for access, the request is denied, with, in some cases, an ABEND code.

  3. For all other environments, RACF issues the information message followed by the operator intervention message. If the system operator gives a negative response to a request for access, the request is denied.

- RACDEF:

  RACF issues an operator message to indicate that RACDEF has been issued and that the request is allowed. Note that, if the user had the ADSP attribute, or PROTECT=YES was specified on JCL for the data set, the resource may be RACF-indicated without a RACF discrete profile being created.

  You can use the operator message or SMF log records at a later time to determine whether the specified resource is in the RACF data set. If it is not, use the ADDSD or RDEFINE command to create a profile for the resource.

The Location, Size, and Number of RACF Data Sets

RACF data sets can reside on any DASD device that is supported by the operating system. Each volume containing a RACF data set should be permanently resident. If RACF is heavily used and you elect to use a single RACF data set, plan to put the data set on a device accessed by a channel and control unit least likely to impact system performance. The use of multiple RACF data sets is recommended to reduce device contention and to reduce the number of resources made unavailable by the loss of one data set or device. The optimum number of RACF data sets for your installation depends on the extent to which you use RACF.

The Number of Resident Data Blocks

The use of resident data blocks reduces the number of I/O requests made to the RACF data set. See *SPL: RACF* for details.

The Size of the SMF Data Set

The amount of space required on the SMF data set depends on the logging options you select for each RACF-protected resource and the average length of your resource names. For estimating storage on the SMF data set, see *System Management Facilities (SMF).*

Routines or Procedures to Extract SMF Records

You might want to modify your existing analysis or reporting routines that process SMF records in order to handle the RACF processing and initialization records, types 80 and 81, and RACF-related information in records 20 and 30. The full formats of records 80 and 81 are in the publication *SPL: RACF.* For record 20 and 30 information and instructions on writing report programs for SMF records, see *System Management Facilities (SMF).*

The RACF report writer provides a standard RACF function for processing the RACF SMF records. The report writer also has installation exit tailoring facilities. For a description of the report writer, see *SPL: RACF.*

Using CLISTs to Define Users

You can use TSO CLISTs to help the migration to RACF.

For example, you can use a CLIST to create a list of the TSO users who are defined in the TSO UADS data set (via the ACCOUNT command). Direct the output of the CLIST to a data set. Then use this data set to select the users you want to protect with RACF. Create a second CLIST containing the selected userids preceded and followed by symbolic variables. Define the users to RACF by executing this CLIST and substituting the preceding symbolic variable with the character string ADDUSER and the following symbolic variables with the desired user operands.

JCL Changes

RACF-defined users must identify themselves to the system as RACF users. When submitting batch jobs, the user can put this information on the JCL JOB statements using the USER, GROUP (optional), and PASSWORD parameters. (A TSO SUBMIT exit routine, TSO/E, or other procedures for handling batch jobs can also place the RACF parameters on the JOB statement.) See *JCL* for details of the parameters.

Note that JES will suppress printing passwords in output listings.

Because the JOB statements contain RACF passwords, you should also establish procedures to ensure the security of the JOB statements. For example, if the passwords are in a card deck, users can put passwords on a continuation statement and use non-printing punches so that the passwords are not easily visible.

RACF-defined users can also use the keyword parameter PROTECT on the JCL DD statement. When the PROTECT parameter is specified, DASD data sets and tape volumes are automatically defined to RACF when the DD statement defines a new DASD data set or a tape volume. See *JCL* for details of how to code the parameter.

Note: If the data set you are defining is adequately covered by a generic profile, do not use the PROTECT parameter because this will force the creation of a discrete profile.

TSO Changes

In order for users to log on to TSO, the users must have TSO profiles (UADS data set entries).

TSO profiles indicate whether or not users can have volumes mounted during a TSO session. You may want to give this ability to users (including IBMUSER) who are going to be defining resources to RACF using the RACF ADDSD or RDEFINE command.

RACF users can use two RACF-related parameters on the TSO LOGON command: NEWPASSWORD (to specify a new password to replace the current password) and GROUP (to specify the group name to which the user is connected during the terminal session). See the *TSO Command Language Reference* for details of the parameters. In addition, a user with the OIDCARD attribute must supply a valid operator identification card during logon.

Note: Although the TSO/Extensions (5665-285) is no longer a prerequisite for RACF, it must still be installed if you wish to use SUBMIT command parameters for batch jobs that access RACF-protected resources. Also, TSO/E Release 2 is required in order to use the ISPF panels.

Maintaining RACF Data Sets

As an alternative to maintaining all your RACF profiles on one data set, RACF offers multiple RACF data sets. With up to 255 RACF data sets, you can spread accesses across multiple devices, thus reducing device contention and also reducing the number of resources made unavailable by the loss of one data set or device.

Note that the RACF data sets should, themselves, be RACF-protected.

See *SPL: RACF* for details on how to maintain RACF data sets, switch to alternate RACF data sets, coordinate profile updates, and make other considerations when dealing with shared or multiple RACF data sets.

Chapter 7. RACF and IMS/VS

This chapter describes factors IMS/VS owners and administrators should consider when using IMS/VS and RACF. Note that many of the names used in the scenarios are arbitrary. The names you use for userids, groupnames, and other such items will differ, and the procedures you decide to follow may vary from those given as examples here.

RACF/IMS Overview

RACF can be used by IMS/VS 1.1.5 and later IMS/VS releases when running under the MVS operating system.

IMS/VS uses RACF facilities to:

- Provide individual user (terminal operator) identification and authentication
- Control access through operator identification:
  - control access to IMS/VS physical terminals
  - Control access to IMS/VS transactions
  - Control access to IMS/VS control regions
  - Control access to control region resources (PSBs, transactions, and logical terminal names) for message processing regions and batch message processing regions
- Delegate the ability to define users, protected transactions, and transaction lists to people outside the data processing department, if so desired
- Provide an audit trail to the individual operator on the IMS/VS log
- Provide an operator identifier to application programs (through the IO/PCB)

You should be familiar with the IMS/VS system generation process and the other security features of IMS/VS, such as password, logical terminal, and transaction security.

The scenarios in this chapter describe how to use RACF to control access to IMS/VS resources. The order of the scenarios is, generally, from the easiest and most cost-effective to the more complex. The scenarios are:

- Controlling access to IMS/VS system data sets and data bases
- General system generation considerations
- Establishing audit trail capabilities
- Controlling access to IMS/VS control regions
- Controlling access to IMS/VS transactions
- Controlling access to IMS/VS physical terminals
- Controlling access to IMS/VS control region resources by dependent regions

Controlling Access to IMS/VS System Data Sets and Data Bases

As described earlier in this publication, RACF can control access to resources such as data sets, tape volumes, transactions, and so forth. Many of the IMS/VS system resources fall into the data set class. Examples of these resources are the data sets that comprise the IMS/VS data bases and libraries.

If access to the IMS/VS data base and library resources is not controlled, further access control within IMS/VS is of little value because the controls within IMS/VS depend upon these resources.

It is reasonable, then, to use RACF to control access to these data sets prior to using RACF to control access to resources such as transactions or terminals. In doing this, it is important to understand that many of the accesses to IMS/VS data sets and libraries are made by IMS/VS itself. That is, to RACF, the IMS/VS control region is a user. If IMS/VS runs as a started procedure, you must place the procedure's name (usually IMS) in the RACF started procedures table (RACF module name ICHRIN03), unless the started procedures table has already been modified at your installation to contain a generic entry. For more information on loading a RACF started procedures table, see *SPL: RACF.*

Before the security or group administrator who has IMS responsibility defines the IMS/VS control region in the started task table, it is a good idea for this person to define IMS as a user to RACF.

```
ADDGROUP IMSVS  OWNER(imsadmin)
ADDUSER IMS  GROUP(IMSVS)  OWNER(imsadmin)
```

These two RACF commands build a RACF profile for a group named IMSVS and a profile for a user whose identifier is IMS and who is a member of the group IMSVS. In both cases, the OWNER of these profiles is a user called 'imsadmin'. (The owner of a RACF profile is allowed to issue other RACF commands that affect the profile. The owner can alter or delete the profile.)

In order to protect the IMS/VS libraries, a person with appropriate RACF authority can issue the following type of RACF commands:

```
ADDSD ('IMSVS.RESLIB','IMSVS.PROCLIB',
                    'IMSVS.ACBLIB', etc...)
       UACC(NONE) AUDIT(ALL) OWNER(imsadmin)
PERMIT 'IMSVS.RESLIB' ID(IMS,.....,other
                                users and groups)
        ACCESS(READ)
PERMIT 'IMSVS.RESLIB'
        ID(users and groups who maintain)
        ACCESS(UPDATE)
etc.
```

A UACC of READ says that anyone can read the data set. A UACC of NONE says that, to do anything with the data set, the individual (or a group to which the individual belongs) must be in the access list for the data set with the appropriate level of permission. In order to prohibit unauthorized parties from opening the IMS/VS system libraries, you should specify NONE as the UACC for these data sets.

IMS/VS libraries (such as RESLIB, PROCLIB, ACBLIB, FORMAT, MATRIX, and JOBS), and IMS/VS system data sets (such as QBLKS, SHMSG, and LGMSG) should be RACF-protected with access granted to 'IMS' (or to the group to which 'IMS' belongs) at the appropriate level, and to the people whose job it is to maintain these libraries and system data sets.

As soon as the ADDSD commands outlined above are executed, the data sets are marked as being RACF-protected. All data management modules in MVS check for the RACF bit in the data set control block. If it is on, they issue the RACHECK macro and supply the name of the data set. RACHECK retrieves the data set profile from the RACF data set and scans the access list, looking for the userid or group name specified in the ACEE for the address space. RACHECK returns a code indicating whether or not access should be allowed, and the data management module allows or denies the access accordingly.

Once access to the IMS/VS system data sets has been controlled to an acceptable level, the next logical step is to control access to the IMS/VS data base data sets. This can be done in the same manner as described for the IMS/VS system data sets or access to these data sets can be controlled with *generic* profiles.

A generic profile provides an access list for a large number of resources (for example, all data sets that have the first two qualifiers of IMSVS.PROD.).

As the appropriate security or group administrator, you can issue RACF commands similar to the following to establish a generic profile:

```
ADDSD 'IMSVS.PROD.*'
      OWNER(imsadmin) AUDIT(ALL)
      UACC(NONE|READ)
PERMIT 'IMSVS.PROD.*'
       ID(IMS,...other users and groups)
       ACCESS(CONTROL)
PERMIT 'IMSVS.PROD.*'
       ID(appropriate users and groups)
       ACCESS(READ)
etc.
```

Issuing the PERMIT command for the user IMS with an access level of CONTROL is necessary because IMS/VS issues a VSAM VERIFY for its VSAM data sets. (VERIFY requires a CONTROL level of access.) For OS data sets, IMS/VS Release 1.2 (and subsequent releases) opens the data set at the highest intent in the processing option for the PSB used. A CONTROL level of access for an OS data set is equivalent to UPDATE.

If your MVS system is not at a level that includes always call in MVS data management modules, you must turn on RACF indicators for the IMS/VS data base data sets that are to be controlled by a generic profile. You can use TSO CLISTs to do this.

Until IMS/VS Release 1.2, batch IMS/VS always opened OS data sets for UPDATE even though the processing intent might be READ. With Release 1.2 of IMS/VS, batch IMS checks the processing intent of the PSB to be used, and, if it is READ, requests an open at the READ level. (This is true for OS data sets only. For VSAM data sets, batch IMS issues the VERIFY macro, which requires a CONTROL access level regardless of the PSB's intent.) This means that people who back up the IMS/VS data bases must still be in the access list of VSAM data sets with a CONTROL level of access, even though they are just reading the data sets.

Note: For data set protection to be complete, the catalogs for the IMS/VS system libraries and data sets and the data base data sets must be RACF-protected at the appropriate level.

The steps outlined above provide a great deal of control for a small amount of effort. No IMS/VS system generation or utility runs are needed to achieve this important degree of control.

IMS/VS System Generation Considerations

To have IMS/VS use RACF to control access to the IMS resources, it is necessary to go through an IMS/VS system generation process. Regardless of the type of security chosen, values must be supplied in the IMSCTRL macro and the SECURITY keyword of that macro.

In the IMSCTRL macro, the IMSID= parameter (the name that identifies a specific IMS/VS control region) must be supplied if any form of resource access is to be used. The default value for IMSID= is IMSA. If multiple IMS/VS control regions share a common RACF data set, the IMSID= value must be unique for each control region.

In the SECURITY keyword of the IMSCTRL macro, a parameter value should be supplied for the RCLASS= keyword. The default value supplied by IMS/VS is 'IMS'. IMS/VS uses this value to specify which RACF class name to use to identify the resource classes belonging to a specific IMS/VS control region. For example, if the default (IMS) is taken, then the RACF resource classes for that control region will be:

TIMS for the transaction class
GIMS for the group transaction class
AIMS for the application resource class

If RCLASS=ABC had been specified, then the RACF resource classes for that control region would be:

TABC for the transaction class
GABC for the group transaction class
AABC for the application resource class

IMS/VS uses these class names to perform checks for resource access. For example, with transaction authorization active, IMS/VS issues a FRACHECK macro when a transaction is received. The form of this macro is:

```
FRACHECK ENTITY='resource-name'
         ,CLASS='class-name'
         ,ATTR=READ
         ,ACEE=acee-address
         ,etc.

where 'resource-name' is the name of the
transaction, 'class-name' is TIMS (or TABC
for example), and acee-address is the RACF
control block built during the operator sign
on to IMS/VS.  This control block contains
the USERID AND GROUP-NAMES of the user who
issued the /SIGN ON command.
```

By making these class names unique for each IMS/VS control region, it is possible to have the same transaction code name on a production system and a test system, yet have different access lists for each of them with no ambiguity.

The class names must be defined in IMS/VS as outlined above, and they must also be defined to RACF. The RACF class descriptor table indicates the names of the classes of resources of which RACF is to be aware. The class descriptor table contains three IMS/VS resource classes when RACF is shipped. These classes, each indicated as an entry in the table, are:

- TIMS
- GIMS
- AIMS

The installation must add to this table any other class names it wished to define. See *SPL: RACF*. Note that, while RACF allows up to eight characters in a class name, only the first four characters of a class name are included in the RACF data set profiles. It is therefore recommended that the first three characters of the values placed in the RCLASS parameter of the IMSCTRL macro be unique. (IMS/VS provides the first character of the RACF class name.)

You must activate the IMS/VS resource classes with the RACF SETROPTS command before any IMS/VS resources can be defined to RACF. Activating the IMS/VS classes in RACF has no effect on IMS/VS until options are specified in the IMS/VS system and IMS is started with the options in effect.

Establishing Audit Trail Capabilities

Before you control access to any IMS/VS internal resources, it may be desirable to have some or all of the users of the on-line system learn to identify themselves to the system. This can be done on a voluntary basis; or it can be enforced on the basis of a user group, by physical terminal, or by the entire system.

IMS/VS will use RACF to identify and verify terminal operators who use the /SIGN ON command. The IMS/VS command processor issues a RACF macro called RACINIT and passes a userid, a password (and a new password if one is present), a group-name (if one is present), and a pointer to where IMS/VS wants the accessor control block (ACEE) to be anchored. Other parameters that can be passed to the RACINIT macro are covered later in this chapter.

The RACINIT macro generates a supervisor call that accesses the RACF data set and looks for the specified userid. If a profile is found for that user, RACF checks the supplied password against the password in the user's profile. (Both the password in the user profile and the supplied password are encrypted before the comparison.) If the supplied password is correct, RACF then checks to see if the user's password has expired. If it has, RACF returns a code indicating this to IMS/VS. IMS/VS returns a message to the user indicating that a new password is required.

If the password is correct and has not expired, RACF checks to see if the user is a valid member of the group to which the user signed on (if the GROUP keyword was supplied), or else uses the default group identifier from the user profile. Additional checks are made concerning group membership.

In the normal case, where everything checks out, RACF builds an ACEE and places a pointer to it in the communication terminal block (CTB) for the terminal. The IMS/VS term for this ACEE is the RACF Token. IMS/VS also marks the terminal as being in a signed-on state.

Any time an IMS/VS terminal is signed on, a type X'16' log record is written. This log record contains the physical terminal identifier, the user identifier, and the IMS/VS time-stamp. A X'16' record is also written when the terminal is signed off --either as the result of a /SIGN OFF command or by another /SIGN ON command being entered from that same terminal.

If a terminal is signed on, the user identifier from the RACF Token is placed in several other log records. These include:

- The input message log record (X'01')
- The output message log record (X'03')
- All data base change records (X'50',X'51',X'52')

In addition, the user identifier is made available to the application program in an extension of the IO/PCB.

These facilities enhance the ability to use the IMS/VS log records as an audit trail, whether or not RACF is used to control access to any IMS/VS resource.

To cause IMS/VS to use RACF for a sign on, the only thing you need to do is an IMS/VS system generation with two keywords in the SECURITY macro:

```
SECURITY TYPE=(,RACFTERM)
         ,SECLVL=(,SIGNON)   (or 'FORCSIGN')
```

The RACFTERM keyword specifies that IMS/VS will use the RACINIT macro to identify and verify the user who issues the /SIGN ON command.

SIGNON or FORCSIGN provide the appropriate parameters for the execute statement in the IMS/VS procedure. SIGNON can be overridden by the master terminal operator during /NRE processing to deactivate sign on processing. FORCSIGN indicates that the master terminal operator cannot override sign on processing during an IMS/VS start.

If this is all that is done, terminal operators can optionally issue the /SIGN ON command. If operators have been defined to RACF, and if they provide their password correctly, the /SIGN ON command will be accepted and all of the actions described earlier will take place. (A rejected sign on will result in a DFS2467 message and a X'10' log record will be written.)

Defining a User to RACF: To define a user to RACF, a person with proper authorization issues the following RACF command from TSO:

```
ADDUSER user-id NAME(user-name)
```

This RACF command causes a user profile to be built on the RACF data set with a number of defaults. The user's password is set equal to the definer's group name and is marked 'expired'. The user's default group is set to the definer's logon group. The defined user has no unusual authorities of any kind so far as RACF is concerned. (You can find information on the ADDUSER command in the *RACF Command Language Reference* or by entering 'HELP ADDUSER' from a TSO terminal.)

You can make a user a member of more than one group by using the RACF CONNECT command.

Without doing anything other than just described, a user may sign on to IMS/VS, but is not required to. (Note that any RACF-defined user can use the /SIGN ON command to gain access to a terminal connected to an IMS/VS control region.)

There are several options available to ensure that users sign on to a particular IMS/VS control region. These options are described from the simplest to the more complex:

The easiest method to enforce sign on for all users is to run the IMS/VS security maintenance utility (SMU) with the following:

```
)(SIGN
  STERM ALL
```

Assuming that the IMS/VS system generation options referenced previously had been done, this run of the security maintenance utility and the EXECUTE options specified by the SECLVL keyword indicate that IMS/VS will pass the terminal identifier of the terminal to the RACINIT macro. This is another of the checks

that the RACINIT SVC will make during sign on processing -- can the user use the terminal specified by the caller of RACINIT?

The return code RACF supplies to IMS/VS indicates whether or not access to the terminal is to be allowed. There are conditions in RACF that determine what this return code will be. The first of these conditions is set by the RACF SETROPTS command:

```
SETROPTS CLASSACT(TERMINAL) TERMINAL(READ)
```

This command sets the TERMINAL class of resource in RACF to an active, system-wide status. All subsystems using RACF to control access to terminals now have terminal checking active when this command is issued. The READ keyword following TERMINAL indicates how RACF is to view terminals that are not defined to RACF. READ indicates that, if RACF cannot find a profile for that terminal, then it is presumed to be unprotected, and access to the terminal is to be allowed.

The security maintenance utility STERM ALL keywords told IMS/VS that all terminals had to have their sign on bit turned on in the CTB before a transaction could be entered from that terminal. (With READ specified for undefined terminals, the IMS/VS terminals do not have to be defined to RACF, but all terminal operators who use that IMS/VS control region must identify themselves through the use of the /SIGN ON command.)

If you want to be selective about which users are to be forced to sign on, see "Controlling Access to IMS/VS Physical Terminals" later in this chapter.

Note that the IMS/VS command processor is not conversational; therefore there is no opportunity to format the screen prior to a sign on. Because the screens on the terminals are not formatted for the /SIGN ON command, the user's passwords are displayed. It is recommended that message format screens be created to provide the sign on function so that the operator might enter /FOR SIGN and receive a formatted screen that requests the password in a non-display field.

Controlling Access to IMS/VS Control Regions

With the steps covered in the prior topics in place, a user must use the /SIGN ON command to enter a transaction from an IMS/VS terminal. The only problem here is that any RACF-defined user could use the /SIGN ON command to gain access to the IMS/VS control region.

To restrict access to a control region to just those users of the system who are authorized by the nature of their jobs to access it, you must set the APPLICATION class in RACF to an active state. You make the APPLICATION class active by issuing the RACF SETROPTS command:

```
SETROPTS CLASSACT(APPL)
```

When the APPLICATION class is active, RACF accepts commands that define application names and their access lists. In the case of IMS/VS, the RACF commands to do this are of the following type;

```
RDEFINE APPL(ims-id1,ims-id2,,,,,,ims-idn) UACC(NONE)
PERMIT ims-id1 CLASS(APPL)
       ID(group-name-1,,,,group-name-n)
       ACCESS(READ)

where ims-idn is the name placed
in the IMSID= parameter of the
IMSCTRL macro.  (The default value
in IMS/VS is 'IMSA').
```

IMS/VS supplies the control region's application identifier to the RACINIT macro during sign on. RACINIT checks to see if the user who is signing on has permission to use the resource pointed to by the APPLID. If no APPLID has been defined to RACF, it is assumed that the application is unprotected. RACF makes this check only when your installation forces sign on.

When IMS/VS forces sign on, all users of a particular IMS/VS control region must be identified and their userids or the names of the groups to which they belong must be in the access list for that particular control region name.

Note that dependent regions are users of IMS/VS control regions. Thus, in order for the dependent region to make the connection to the control region, the JOB statements that start the dependent regions must have a valid userid and a current password for the user. To prevent unauthorized disclosure of the passwords in these JOB statements, a new IMS/VS system data set has been included in the procedures that bring up the IMS/VS control region. This data set is represented by the IMSJOBS DD name. It should be RACF-protected, and the IMS started procedure name should be in the access list with READ access. No one should have access to this data set as a matter of course. When the data set must be changed, RACF commands can be used to allow update access to it. The access authorization can then be revoked after the data set has been updated.

Controlling Access to IMS/VS Transactions

The TRANAUTH parameter, which you specify with the SECLVL keyword of the SECURITY macro, causes IMS/VS to use RACF to control access to IMS/VS security transactions:

```
SECURITY TYPE=(,RACFTERM)
         ,SECLVL=(TRANAUTH,SIGNON)
```

You can also include the TRANAUTH parameter in the initial system generation described earlier in this chapter. As long as the RCLASS parameter agrees with the RACF class descriptor table, the fact that no transaction profiles have been defined does not affect IMS/VS operation.

The RACFTERM parameter tells IMS/VS to use RACF to process the /SIGN ON command. The TRANAUTH parameter tells IMS/VS to do two things:

1. During initialization of the control region, IMS/VS issues the RACLIST macro. RACLIST reads all of the profiles in the specified class (TIMS, GIMS) into a subpool in the local system queue area (LSQA) and arranges an index to them in the form of a binary tree.

2. When a message is received by the communications I/O processor, IMS/VS supplies to FRACHECK the name of the transaction, the transaction class name, and a pointer to the RACF token. FRACHECK is a branch-entered routine that does resource access control checking for the caller. FRACHECK finds the indicated resource class anchor point and locates the resource profile. If the profile is found, FRACHECK checks to see if the user (or the group to which the user signed on) is allowed to access the resource. The return code from FRACHECK indicates the status of the request: access-allowed; access-not-allowed; or profile-not-found. Because IMS/VS treats the profile-not-found return as a not-protected condition, it is perfectly acceptable to have transaction authorization active, but not have any profiles specified for a given control region. The amount of overhead caused by a not-found condition is small.

The use of in-storage profiles has a drawback; refreshing the profiles requires a restart of IMS/VS. This could mean that the tables built by IMS/VS during initialization would not be current under two circumstances:

1. When it is desired to control access to a transaction that previously was unprotected (or a new transaction).

2. When it is desired to change the access list for a given transaction.

Prior to Release 1.3 of IMS/VS it is not possible to identify a new transaction to IMS/VS without going through the restart process. Therefore, the first case does not seem to be too severe. The second case, however, is a concern because work assignments can change frequently, and a person who is not normally allowed access to a transaction might someday be required to access that transaction on either a temporary or permanent basis.

The solution to the second problem is to make sure that the access list for any in-storage profile contains RACF group names rather than individual user identifiers. A user's group affiliation is determined at the time the user signs on, so it is 'refreshable' each time the /SIGN ON command is entered. A user's access to a particular resource can be granted by making the user a member of the appropriate group.

Assume a given transaction has in its access list the name of a given RACF group--say GROUP1. Also assume that a given user (Sally) needs to be able to use that transaction and that Sally is not in the access list, nor is she a member of GROUP1.

The owner of GROUP1 (or someone with at least CONNECT authority in that group) can make Sally a member of that group by issuing the RACF command:

```
CONNECT Sally GROUP(group1)
```

When Sally signs on to IMS, she includes the GROUP(GROUP1) keyword in her sign on. She can then access anything that GROUP1 is allowed to access. To prevent Sally from accessing the transaction at some later time, the owner of GROUP1 issues REMOVE Sally, and she can no longer sign on as a member of GROUP1.

It is recommended that the group structure defined to RACF reflect business functions at the lowest level of group. A business function would be something like Order Entry, Customer Inquiry, New Customers, Name and Address Changes, and so on. Supervisors of each of these business functions could be given the RACF authority to connect defined users to their groups (business functions). Thus, the supervisor of the order entry function would control who could do that function. In turn, the owner of the order entry subsystem could control what business functions could access the transactions that make up that function. Release 1.3 of IMS/VS provides the facility to refresh the RACF in-storage profiles through the use of an IMS/VS command.

Because of the way IMS/VS uses RACF, users are not the only entities that can be grouped under a single name. IMS/VS transactions can also be grouped so as to have a common profile. Assuming that the order entry process requires the use of four different IMS/VS transactions, they could all be defined to RACF at once and all could share a common access list:

```
RDEFINE Gxxx(ORDENT) ADDMEM(TRAN1,TRAN2,TRAN3,TRAN4)
       UACC(NONE)
PERMIT ORDENT CLASS(Gxxx) ID(GROUP1) ACCESS(READ)
```

Execution of these two RACF commands is the same as issuing RDEFINE and PERMIT commands for each of the four transactions, but there is no need to define the transactions individually. The RACLIST macro builds an entry for each transaction and points to the common profile as if each one had been defined individually, and storage is saved because only the transaction names are unique. They share a common profile containing all other RACF information such as ownership, UACC, statistics, and the access list.

You can, of course, have an individual profile for a transaction that is already defined in the Gxxx class. If both a group profile and an individual profile exist, RACLIST merges the two profiles and uses the merged profile. If there are conflicting specifications in the two profiles, RACLIST resolves these through a set of rules that can be specified by flags in RACF.

Once you have set the IMS/VS classes active with the RACF SETROPTS command, and the IMS/VS start up procedure specifies TRANAUTH, the transactions defined to RACF at that time will be RACF-protected. IMS/VS issues the FRACHECK macro at four different times:

1. Before placing a transaction in the scheduler message block
2. When a 'CHNG' call is issued to a modifiable IO/PCB
3. When an 'ISRT' call to a scratch pad area contains a transaction name
4. When the /SET, /LOCK, and /UNLOCK commands contain transaction names

Note that the effect of the checking calls on a transaction-to-transaction switch. A terminal operator could sign on and enter a transaction that may or may not be RACF protected. If the transaction were to stay in the queue long enough to be scheduled after the operator has signed off, there would no longer be a valid RACF token associated with the terminal. If the first transaction then invoked a protected transaction through a CHNG call to a modifiable IO/PCB, the scheduling of the protected transaction could fail.

IMS/VS will use RACF to force reverification that the operator who signed on to a given terminal is the same one who is entering a second or subsequent transaction. This is done by including the word 'REVERIFY' in the APPLDATA field of the transaction profile:

```
RDEFINE Txxx(tran-name) UACC(NONE) APPLDATA('REVERIFY')
```

Each time users enter this transaction code, they must enter their RACF password in the place where an IMS/VS password would go if the transaction were password-protected.

Use of this method prevents the problems associated with terminal operators leaving their terminals in a signed on state. However, it has adverse implications on message formats and on usability and productivity. It would be better if supervisors impressed on the terminal operators the importance of not leaving their terminals in a signed-on state. The transaction authorization exit in IMS/VS could be used to do elapsed time checking between transactions, but even this does not assure that a transaction can not be invoked by a person other than the one who signed on and left the terminal unattended.

Note that generic names can be used in the transaction class (Txxx) or as members in the group class (Gxxx) to cover multiple transactions with similar names and authority requirements.

Controlling Access to IMS/VS Physical Terminals

If you want to treat IMS/VS terminals as resources (that is, to allow only certain users or groups of users to use physical terminals connected to IMS), you must describe the terminals to RACF and build access lists for them.

If a terminal is defined to RACF and the resource manager supplies the terminal identifier to RACF during RACINIT processing, RACF returns a code that indicates whether or not this user is allowed to use this particular terminal. As was indicated earlier, the signal to IMS/VS to supply the terminal identifier comes from the security maintenance utility.

```
)(SIGN
   STERM node-name      (VTAM,TCAM)
   STERM lll#ttt        (BTAM relative line and term #)
```

Entries like the above cause IMS/VS to pass the terminal identifier to RACINIT during /SIGN ON processing.

If the terminal is defined to RACF, the access list for that terminal is scanned and RACF indicates to the sign on command processor whether or not terminal access is allowed.

If a terminal is not defined to RACF, the code returned by RACF depends on the UACC allowed for undefined terminals. The SETROPTS CLASSACT(TERMINAL(READ)) command says that undefined terminals can be accessed by anyone. However, there is a group-related attribute that can override this specification.

When you define a RACF group profile (or change one with the ALTGROUP command), you can tell RACF whether or not the undefined terminal UACC is to apply to this group. You do this by using the TERMUACC | NOTERMUACC keyword in the RACF commands. (The default is TERMUACC.) TERMUACC indicates that, for this group, the system-wide level of access is to be used for undefined terminals. NOTERMUACC indicates to RACF that, for this group, RACF is to ignore the system-wide undefined terminal access level of READ and to prevent this group from signing on to undefined terminals. In other words, members of this group must be specifically allowed access to the terminals they want to use.

As in the case of transaction authorization, unless you restart IMS, you cannot change the IMS/VS security matrix indicating the terminals to be RACF protected. (This is true for all releases of IMS prior to Release 1.3. With Release 1.3 of IMS/VS, you can reload the security matrix without restarting IMS. This is the same facility that allows the RACF transaction authorization profiles to be reloaded without restarting the system.)

Controlling Access to IMS/VS Control Region Resources by Dependent Regions

IMS/VS control region resources, such as program specification blocks, transaction names, and logical terminal names, are accessible to programs operating in dependent regions. To MVS, these dependent regions are normal MVS jobs, and they can be initiated through the MVS job entry subsystem by anyone. Thus, a person not authorized to access a data base through a RACF-protected IMS/VS transaction could access it through a batch message processing region by putting the proper parameters in the EXECUTE statement for the BMP.

The application authorization security or resource access security facility of IMS/VS can use RACF to provide additional control of who can access the resources of the control region through dependent regions. The vehicle through which this is done in IMS/VS is the application group name (AGN).

Application group names are defined to IMS/VS through the security maintenance utility. They can be used with or without RACF. If RACF is not used, then you must employ an installation exit routine to authorize access to control region resources by dependent regions. The installation exit routine supplied by IBM will deny all access with resource access control active.

When IMS/VS resource access security is implemented using RACF, both IMS/VS and RACF must be aware of the resource names. To make RACF aware of

resource access security, activate the Axxx class of resource in RACF with the SETROPTS command:

```
SETROPTS CLASSACT(Axxx)
```

```
where Axxx is the name of the IMS/VS resource
class derived from the RCLASS=xxx keyword of
the IMSCTRL macro.  The default value is AIMS.
(Be sure this name is in the RACF class
descriptor table.)
```

With the Axxx class active in RACF, you can define application group resource names to RACF without any impact on IMS/VS. IMS/VS will not use this facility until told to do so through the system generation process and a start of IMS/VS.

To define the application group names to RACF, use the RACF RDEFINE and PERMIT commands as with other resources:

```
RDEFINE Axxx(agname-1,,,,,,,agname-n)
        OWNER(imsadmin) UACC(NONE)
PERMIT agname-1 CLASS(Axxx)
       ID(group-1,,,,group-n) ACCESS(READ)
etc.

RACF has no idea of what these names
represent; they are just names in a
given class.  Each name has a list of
the users or groups of users who have
been allowed to use the name.
```

To cause IMS/VS to issue the RACHECK macro during dependent region initialization, you must code the RACFAGN keyword in the SECURITY macro for the IMS/VS system generation. You code it in the TYPE=(RACFAGN,,) statement. There is no keyword to specify control of how the master terminal operator may override this option (as there is, for example, in other options in the SECLVL statement.) The RACFAGN statement causes the inclusion of the call to RACF and conditions the job control language for the control region to have the value ISIS=1 in the EXECUTE statement for it. The ISIS=1 specification indicates that RACF is to be used for resource access security. ISIS=2 says use the resource access exit routine written by the installation. ISIS=0 says do not use resource access security. Once again, these parameters cannot be overridden by the master terminal operator during a restart of the system. They can be affected by the RGSUF parameter in the IMS/VS EXECUTE statement, however.

Before activating resource access security with RACF, it is important to note that all dependent region job control language will be affected by this feature. All job statements must contain valid RACF-defined user identifiers and current passwords. (You can define these users to RACF with passwords that never expire, if you desire.) For this reason, IMS/VS now has a new DDNAME, IMSJOBS, whose data set should be RACF-protected with a UACC of NONE, and the IMS started task user identifier in the access list should have at least READ permission.

In addition to this, the EXECUTE statements for the dependent regions must specify the IMSID of the control region (IMSCTRL macro), the application group name (defined to RACF and in the security matrix), and, for batch message processing regions, the names of the resources to be accessed by this region.

You must run the security maintenance utility to assign control region resource names to a given application group name:

```
)(AGN  agname-1
   AGPSB psb-1
   AGPSB psb-2
   AGPSB psb-n          >Example for batch message processing (BMP)
   AGTRAN tran-1
   AGTRAN tran-n
   AGLTERM lterm-1
   AGLTERM lterm-n

)(AGN  agname-2
   AGTRAN ALL           >Example for message processing (MPP)
   AGLTERM lterm-n
```

For message processing regions, the EXECUTE parameters include only the control region name and the application group name. If a message processing region tries to schedule a transaction that is not in the application group name for that region, the scheduling will fail.

For batch message processing regions, the EXECUTE parameters include the control region name, the application group name, and the names of the transaction, the program specification block, and the logical terminal that the program wishes to use.

The checking process is a two-step process. The first check involves RACF; the second does not.

At dependent region initialization time, the MVS initiator issues the RACINIT macro and builds an accessor control environment element (ACEE) for the address space. This occurs for any address space initialization. When IMS/VS "connects" a dependent region to the control region, the IMS/VS module that does this (DFSSCHRO) is running under the task control block of the dependent region. This module performs the RACHECK function and supplies the RACF class name (Axxx) and the name of the AGN passed to it in the EXEC statement parameter list. If the user is not allowed to use the application group name, RACF returns a "bad" condition code and IMS/VS does not allow the dependent region to connect. If RACF returns a "good" condition code, the connection is made.

The second part of the two-step process is an IMS/VS function only. IMS/VS checks the name of the transaction or PSB or logical terminal being requested by the dependent region against the entries in the security matrix and allows or disallows use depending on whether or not the name is in the entry for the application group name.

For message processing regions, application resource security is somewhat like class scheduling in that transactions can be scheduled only in regions whose application group name allows them.

For batch message processing regions, this level of control prevents unauthorized users from starting an MVS job that can access the resources defined in the control region.

Summary

You can enhance the security and integrity features of IMS/VS to a significant degree by using RACF in the MVS environment.

Any security mechanism is only as good as the management control of the people who use the system. IMS/VS and RACF provide the tools to enhance control of a critical resource. It is management's responsibility to see that the controls that are implemented are working the way they are supposed to work, and that variances are reported to and acted upon by management.

In this regard, RACF, with its lists of users and lists of resources, allows management to delegate the authority to the owners of these entities in such a way as to maintain the separation of duties while maintaining a flexible, responsive access control strategy.

In order to be effective, access control must allow management to adopt the principle of "least possible privilege" for those resources that are deemed to be highly sensitive. This principle says that access to these resources is controlled in such a way that permission to use them is restricted to just those people whose normal duties require their use. Any unusual use of the resource should be approved by an administrator or manager, as well as the owner of the resource.

The delegation mechanism in RACF and the easy, non-technical commands that change the relationship of a user to a resource mean that adopting the principle of "least possible privilege" need not be burdensome nor inflexible when unusual circumstances dictate that access permission should be changed. When an unforeseen circumstance requires a change in access privilege, the change can be made by a non-technical person with access to a TSO terminal, and management can be alerted to review the fact that the change was made.

Through the use of RACF, the security functions of IMS/VS can move out of the highly centralized environment required previously and into a more flexible, responsive, and secure environment.

Chapter 8. RACF and CICS/VS

This chapter describes steps to consider when using RACF with CICS/VS. Note that the scenarios do not represent the **only** way to prepare your CICS/VS system to use RACF; the procedures you decide to follow may vary from those given as example in this chapter.

RACF/CICS Overview

RACF can be used by CICS/VS Release 1.5 in an MVS environment to:

- Control access to CICS/VS transactions in an online environment
- Control access to Data Language 1 (IMS/DB) program specification blocks from batch regions

Release 1.6 of CICS/VS provides these two functions and also extends the use of RACF to:

- Control access to CICS/VS online regions
- Control access to the CICS/VS operator terminals
- Control access to program specification blocks (PSBs) by online terminal operators

Preparing CICS/VS for the Use of RACF

The remainder of this chapter describes a number of steps you can take to apply RACF protection to your CICS/VS system. The order of information is, generally, from the easiest and most cost-effective to the more complex. A summary section follows the steps. You should be familiar with the CICS/VS system generation process and the security features of CICS/VS, as well as with RACF.

The steps are:

- Controlling access to the CICS/VS system libraries
- Controlling access to the CICS/VS program libraries
- Controlling access to the CICS/VS files and data bases
- Preparing CICS/VS to use RACF
- Identifying all users of the CICS/VS online system
- Protecting sensitive CICS/VS transactions

- Protecting DL/1 program specification blocks
- Controlling access to CICS/VS regions (CICS/VS Release 1.6 only)
- Controlling access to CICS/VS terminals (CICS/VS Release 1.6 only)
- Defining "remote" accessors of CICS/VS resources

Controlling Access to the CICS/VS System Libraries

Because the CICS/VS system libraries control how the rest of CICS/VS works, it is reasonable to control access (especially access greater than the READ level access) to these libraries. That is, you should RACF-protect data sets in the CICS/VS system libraries with a UACC of READ or NONE. You should restrict access authority of UPDATE or higher to those people whose job descriptions make them responsible for the contents of the system libraries.

When CICS/VS runs as a job, the only thing the security or group administrator need do to provide basic system library security is to include the USER=,PASSWORD= parameters in the JOB statement that starts CICS/VS, and give READ access (to the data sets in the system libraries) to the CICS userid.

When CICS/VS runs as a started task, there is no associated job statement. The task name and the associated userid should be inserted into the RACF started procedure table. See "Using Started Procedures" in Chapter 5 for details. See also *SPL: RACF.*

Note that some of the CICS/VS system data sets, such as journal data sets, require that you give UPDATE access authority to the CICS/VS userid for those specific data sets. (You should also give appropriate authority to people who do volume backups on system data sets.)

The userid appearing on the CICS/VS JOB statement must be defined to RACF with a valid, active password, and READ access to the data set containing the JOB statement must be restricted.

Controlling Access to CICS/VS Program Libraries

The programs in the CICS/VS program libraries control what is done by the transactions that are entered by the terminal operators. It is reasonable, then, to control what goes into the program libraries. You can RACF-protect the data sets in the program libraries with a UACC of READ or NONE. If the UACC is NONE, then the access lists for the data sets must include the userid from the CICS/VS JOB statement (because CICS/VS opens these data sets during initialization).

Controlling Access to CICS/VS Files and Data Bases

Unless your computing system is used exclusively by CICS/VS, the online CICS/VS files and data bases are vulnerable to batch, RJE, and TSO users when the CICS/VS system is inactive. (If the disposition of the CICS/VS data sets is SHARE, the data sets are vulnerable all the time.)

For this reason, it is reasonable to restrict UPDATE access authority for the CICS/VS data sets to the CICS/VS region. If it becomes necessary to permit

anyone else to have UPDATE authority, you can do this by delegating the use of the RACF PERMIT command for these data sets.

Preparing CICS/VS to Use RACF

Several CICS/VS macros must be coded to tell the CICS/VS resource management modules to use RACF for their access control functions. These include:

- DFHSG
- DFHSIT
- DFHSNT
- DFHPCT

Include the DFHSG PROGRAM=CSS in the system generation. This causes the following set of programs to be included in the system:

- DFHSNP The sign on transaction
- DFHSFP The sign off transaction
- DFHFEP The field engineering terminal test
- DFHACEE The user identification program
- DFHXSP The CICS security program
- DFHXSS The RACF interface

Include the following in the DFHSIT TYPE=CSECT macro:

```
,EXTSEC=YES
,XSP=YES
,XTRAN=xxx
,XPSB=xxx
,APPLID=
```

The EXTSEC=YES keyword requests checking by an external security manager (that is, RACF).

The XSP=YES keyword indicates that the security program is to be used for resource access control. That is, XSP=NO indicates bypass security checking.

The XTRAN=xxx value, prefixed by a 'T' and a 'G' will be used as RACF class names. CICS/VS transaction names are in the RACF class Txxx, and groups of transaction names are in the class Gxxx. These names must agree with the names in the RACF class descriptor table. The default value in CICS/VS is 'CICSTRN', which results in the default RACF class names of 'TCICSTRN' and 'GCICSTRN'.

The class names Txxx, Gxxx, Pxxx, and Qxxx generated from the XTRAN and XPSB parameters must be defined in the RACF class descriptor table using the ICHRRCDE macro. See *SPL: RACF* for details.

The XPSB=xxx value is used to define the RACF class names for DL/I program specification blocks. It is used in the same way as the value given for XTRAN. That is, the value entered here is prefixed with a 'P' and a 'Q' to result in the RACF class names for PSBs and groups of PSBs. The CICS/VS default name of 'CICSPSB' results in the RACF class names of 'PCICSPSB' and 'QCICSPSB'.

Note: These class names should be defined to RACF even if PSB checking is not required. With Release 1.6 of CICS/VS, this parameter applies to both batch and online PSBs.

The APPLID= value is the identification of this CICS/VS region for the inter-system communication feature, the multiple region option, and the shared data facility. In CICS/VS Release 1.6, this value is the RACF name that is used to control access to the CICS/VS region itself. If the APPLID is supplied in this macro, rather than in the DFHTCT TYPE=INITIAL specification, CICS/VS uses this value to determine whether a user will be permitted to sign onto this CICS/VS region. If the value is left blank, APPLID checking does not take place during sign on.

Identifying All Users of the CICS/VS Online System

In many cases, security and administrative problems can be attributed to users who are not known to the system. Therefore, good security practice dictates that users of valuable resources, in particular, be identified. By using RACF, it is relatively easy to have your operators specify and maintain their own passwords, rather than attempting to keep the operator passwords in the CICS/VS sign on table. Sign-on is required to set a security level higher than the default CICS level of 1 or when the transactions to be used require RACF checking.

To identify terminal operators, it is not necessary to use the CICS/VS sign on table; however, to use CICS/VS security facilities (such as program control, journal control, OPID, and similar CICS/VS facilities) that do not currently use RACF services, you must retain the sign-on table.

When the CICS/VS sign on transaction is executed for a RACF-defined user, CICS/VS issues the RACINIT macro and supplies the user's identifier (1-7 characters), the RACF password (and a new one, if entered), the terminal identifier (CICS/VS Release 1.6), and the CICS/VS region's APPLID (CICS/VS Release 1.6) if the APPLID was supplied in the DFHSIT macro.

RACINIT checks to see whether the user:

- Is defined to RACF
- Knows the correct password
- Has an expired password
- Has been revoked from the system
- Is allowed access to this terminal
- Is allowed access to this CICS/VS region

If the user passes these RACINIT checks, RACF builds an ACEE and places a pointer to it in the terminal control table extension. RACINIT then passes a return code of zero to CICS/VS. The ACEE, which is the control block pointed to by CICS/VS when CICS/VS asks whether a user is allowed to use a requested resource, contains the userid, the RACF group(s) to which the user belongs, and other data about the user.

If the CICS/VS sign on table is not to be used, one entry is required for it:

```
DFHSNT TYPE=(entry,default)
             ,EXTSEC=YES
```

The DFHSNT specification creates one entry in the sign on table with a blank operator name and indicates that the external security facility is to be used for any operator name not found in the sign on table prior to this entry. Because this is the only entry, the sign on function for all operators will use RACF to identify and verify the user. Note that, if you want to use the sign on function selectively, or retain other CICS/VS security functions, you can make individual entries in the CICS/VS sign on table.

The DFHSNT TYPE=INITIAL macro can be used to establish default values for other entries in the sign on table. For example, DFHSNT TYPE=INITIAL,EXTSEC=YES | NO establishes the default for all DFHSNT TYPE=entry statements following it. (The default for EXTSEC=YES or NO is NO. It can be overridden in any subsequent TYPE=entry statements.)

In any event, it is the EXTSEC=YES specification in the sign on table entry for a given operator that causes CICS/VS to call RACF to employ the operator identification and verification function:

```
DFHSNT TYPE=entry
  ,OPIDENT=ooo            The value used for
                          CICS/VS logging.

  ,OPNAME=nnnnnnn         If RACF is being used,
                          the OPNAME field should
                          be 7 characters or less.

  ,EXTSEC=YES             Causes the entry to be
                          checked by RACF.

  ,PASSWORD=pppppppp      The CICS/VS (not RACF)
                          password. It is used
                          only if RACF is inactive.

  ,RSLKEY=n1,n2,...nn     The CICS/VS (not RACF)
                          resource security level
                          key for command level
                          resource security.

  ,SCTKEY=n1,n2,...nn     The CICS/VS (not RACF)
                          transaction security
                          level key. It is used
                          if RACF is not active.
```

In effect, the EXTSEC=YES keyword tells CICS/VS to use RACF for password checking, rather than using the password in the sign on table.

You can have a mixture of operators: some defined in the sign on table, and some not defined. Those defined in the sign on table with EXTSEC=YES will be checked by RACF. Operators who are not defined in the sign on table will be checked by RACF if the last entry in the sign on table is DFHSNT TYPE=(entry,default). Note that at this point, terminal operators can sign onto the system, but they are not required to do so. Also, with the DFHSNT TYPE=(entry,default) specification, any RACF-defined user can sign onto the CICS/VS system.

Protecting Sensitive CICS/VS Transactions

To use RACF checking when CICS/VS transactions are invoked, it is first necessary to define the transactions to RACF, establish access lists for the transactions, and indicate to CICS/VS that access to the transactions is to be controlled through the use of RACF.

The first step is to build profiles in the RACF data set. Use the RACF SETROPTS command to activate the four CICS/VS resource classes so that RACF will recognize them when they are used as objects of RACF commands. (Although the examples use discrete profiles, generic profiles can be used as an alternative in the transaction class (Txxx) or as members in the group class (Gxxx) to cover multiple transactions with similar names and authority requirements. Likewise, generic profiles could alternatively be used in the PSB class (Pxxx) or as members in the group class (Qxxx) to cover multiple PSBs with similar names and authority requirements.)

Next, define the transactions to RACF:

```
RDEFINE GCICSTRN uuuuuuu.gggg
     ADDMEM(uuuuuuu.ttt1,...,uuuuuuu.tttn)
     UACC(NONE)
```

Or, to define individual transactions, specify:

```
RDEFINE TCICSTRN(uuuuuuu.ttt1,...,uuuuuuu.tttn)
     UACC(NONE)
```

Then, to permit operators to access the protected transactions, specify:

```
PERMIT uuuuuuu.tttn CLASS(GCICSTRN)
     ID(opr1,...oprn, group1,...groupn) ACCESS(READ)
```

The CICS/VS transaction resource names are prefixed with the userid from the JOB statement that starts this CICS/VS region.

This completes the resource definition for transactions. The necessary profiles have been built in the RACF data set, but no RACF processing will yet take place (because the CICS/VS system has not yet been told to use RACF.) Note: Before going on, it is a good idea for the definer of a RACF resource to check what has been done by using the RLIST RACF command to view the profile that has just been created.

Because of the way in which CICS/VS handles some RACF return codes, it is necessary to define to RACF all DL/I PSBs accessed by batch regions before you bring up the CICS/VS system with RACF active. (With CICS/VS Release 1.6, all PSBs must be defined to RACF whether or not access to them is to be controlled.) The simplest way to do this is with the RACF RDEFINE command:

```
RDEFINE QCICSPSB(uuuuuuu.public)
          ADDMEM(uuuuuuu.pppppppp,...)
          UACC(READ)
```

Specifying the RDEFINE command in this way has the effect of granting access to all the PSBs defined in the ADDMEM list.

To control access to transactions, tell CICS/VS which transaction invocation is to call RACF through the external security manager. You do this in the CICS/VS definition of the transaction:

```
DFHPCT TYPE=entry
       ,TRANSID=tttt
       ,EXTSEC=YES
```

This "marks" the CICS/VS transactions as those using the external security manager for access control.

Having completed the prior steps, you can now initialize CICS/VS with the system initialization table that indicates:

```
EXTSEC=YES
XSP=YES
XTRAN=xxx
XPSB=xxx
APPLID=aaaa
```

When CICS/VS is being initialized for use with RACF, it requests the RACF RACLIST service and specifies the class names for which RACLIST is desired (in this case, the transaction and PSB classes). From the RACF data set, RACF builds in-storage profiles for all of the resources in the named class. The RACLIST call for the transaction class first builds entries for those transaction codes defined in the group class (GCICSTRN), and then builds entries for the transaction codes in the transaction class (TCICSTRN). If a transaction code was named in both classes, RACLIST merges the information from the two profiles to build a single profile. Conflicts between the two profiles are resolved by rules flags.

Because of the way in which the profiles are built, changes made to the profiles after a CICS/VS initialization do not become effective until the **next** start of CICS/VS. However, operators can still be allowed access to transactions through group membership (group affiliation is determined each time an operator signs on). That is, as long as there is a group name in the access list for a given transaction, access to the transaction can be granted to an operator between CICS/VS initializations by using the RACF CONNECT command to put the operator in a group that can access the resource.

When CICS/VS gets a message indicating that a specific transaction is to be invoked, it checks to see whether the transaction has been marked for external security. If it has, CICS/VS calls its external security manager and invokes the FRACHECK RACF service routine. CICS/VS passes to FRACHECK a parameter list containing the name of the resource (in this case, uuuuuuu.tttt), the resource class, and a pointer to the ACEE built for this terminal when the operator signed on. The FRACHECK routine finds the in-storage profile associated with the transaction name and checks to see whether the userid or the groupid for that operator is in the access list for the transaction. FRACHECK then passes an appropriate return code back to the CICS/VS external security manager. Depending on the return code, access is either allowed or denied. (Note that, if the terminal operator had not signed on, access would be denied, a security message would be sent to the terminal, and a record of the violation would be written to a transient data destination of CICS/VS.

Protecting DL/I Program Specification Blocks (PSBs)

RACF checking of PSBs is not selective. That is, FRACHECK is always issued, and, if the profile for a PSB is not found, access is denied. This applies to PSBs accessed from batch regions through the IRC feature for CICS/VS 1.5 and to both batch and on-line access in CICS/VS 1.6.

To specify those PSBs that can be accessed by anyone, use the Qcicspsb RACF class to define all of them at once:

```
RDEFINE Qcicspsb(uuuuuuu.public)
        ADDMEM(uuuuuuu.psbname1...,uuuuuuu.psbnamen)
        UACC(READ)
```

The UACC of READ allows anyone to use a PSB included in the list of names following the ADDMEM keyword. It is not necessary to define each PSB name individually in the PCICSPSB class.

To define those PSBs to which you want to restrict access, define the PSBs (in either PSB class) with a UACC of NONE, and issue the RACF PERMIT command with the appropriate user and group names in the access list.

Note: If DL/I (IMS DB) is present in the system, PSBs must be defined at the same time RACF transaction authorization is implemented.

Controlling Access to CICS/VS Regions (CICS/VS Release 1.6 Only)

To prevent any RACF-defined user from signing onto CICS/VS, Release 1.6 of CICS/VS passes the name of the CICS/VS region to RACF during sign on processing. RACINIT checks the access list of the profile for that CICS/VS region and grants or denies access accordingly.

In order to have CICS/VS provide its identifier to RACF during sign on processing, you must provide the APPLID= keyword of the DFHSIT macro. If APPLID is specified in the DFHTCT macro instead of the DFHSIT macro, application checking does not take place for the CICS/VS region.

If application checking is desired, you must use the RACF SETROPTS command to set the APPL class active in RACF. (Note that this is a system-wide setting. That is, the class may already be active because some other subsystem required it. Similarly, when you activate the APPL class, another subsystem might be affected, although this is unlikely in that RACF treats an undefined APPLID name as if it had a UACC of READ.)

With the APPL class active, define the CICS/VS regions:

```
RDEFINE APPL(cics-applid) UACC(NONE)
PERMIT cics-applid ID(group1...,groupn) ACCESS(READ)
```

When you have issued the two above commands, only authorized users can sign on to this CICS/VS region. Keep in mind that "remote" systems will have to be permitted to "sign on" to this region. (See "Defining Remote Accessors of CICS/VS Resources" later in this chapter for more details.)

Controlling Access to CICS/VS Terminals (CICS/VS Release 1.6 Only)

During sign on processing, CICS/VS always supplies the terminal identifier to RACF. The RACF macro that CICS/VS uses to identify and verify a user and to check a user's authorization to an APPLID also checks to see whether the user is allowed to access the terminal whose identifier was supplied. If you had defined the terminal to RACF, a system-wide RACF option dictates the return code RACF supplies to CICS/VS. If you had set the terminal UACC to READ (with the SETROPTS command), access would be allowed to undefined terminals. However, some subsystem owners may not want to allow access to undefined terminals. You can handle this situation by using the TERMUACC keyword of the ADDGROUP or ALTGROUP RACF commands.

If you have defined a group with the TERMUACC keyword in its profile, the system's universal access level for undefined terminals is in force. That is, if the system-wide universal access for undefined terminals is READ, access to undefined terminals is allowed for this group of users.

If you have defined a group with the NOTERMUACC keyword in its profile, the terminal must have been defined to RACF and the user must be authorized to use the terminal for the sign on to be allowed.

In other words, the NOTERMUACC specification in the group profile says, "Don't pay any attention to the system-wide terminal option for undefined terminals; users in this group must have explicit authority to use any terminal."

To define a CICS/VS terminal to RACF, issue the following RACF command:

```
RDEFINE TERMINAL(netname1...,netnamen) UACC(NONE)
```

For VTAM and TCAM terminals, the terminal name supplied by CICS/VS will be the NETNAME=nnnnnnnn parameter from the DFHTCT TYPE=TERMINAL macro. For BTAM terminals, the terminal name supplied by CICS/VS is the TRMIDNT=xxxx parameter expanded with trailing blanks.

No matter what the terminal type, once a terminal has been defined to RACF, it must have an access list containing appropriate group or user identifiers in order for an operator to sign on to CICS/VS at that terminal. Note that terminal checking will access the RACF data set each time an operator signs on. Therefore, terminal access list changes become effective immediately.

Defining Remote Accessors of CICS/VS Resources

The resources that are controlled by an owning CICS/VS system can be accessed by other CICS/VS regions through intersystem communication or the multi-region option. DL/I PSBs can also be accessed from a batch region running the IRCBCH facilities of DFHDRP. If the resources in the owning system are RACF-protected, access to them is denied unless CICS/VS supplies a valid userid or group name for FRACHECK processing.

There are two different ways for a remote region to access resources that are owned by another region: through the CICS/VS supervisor call (such as from IRCBCH or MRO), or through intersystem communication (ISC).

If the connecting path is through the CICS/VS supervisor call, the name supplied to the owning system's checking routines is the RACF userid from the JOB statement that started the remote region. This, then, is the name that must be on the access list of the resource in the owning system.

If the connecting path between the two systems is through a telecommunications access method, the name that the owning system supplies to the checking routines is the name specified in the XSNAME= parameter in the owning system's DSFHTCT TYPE=SYSTEM macro that describes the remote system.

For the telecommunications-connected systems, the connection is checked by the telecommunications access method. This is done by matching the APPLID= value from the remote system with the name specified by the NETNAME= parameter of the DFHTCT TYPE=SYSTEM macro in the owning system. The owning system then supplies the value given in the XSNAME= parameter associated with this DFHTCT TYPE=SYSTEM macro as the userid for the sign on of the remote system.

For the sake of consistency, it is good practice to use the same name for the APPLID= parameter of the DFHSIT macro as is used in the USER= parameter of the JOB statement that starts the remote system.

Summary

This section summarizes the steps described in this chapter. The following order of steps allows the RACF setup work to be done before the CICS/VS system generation and initialization processes:

In RACF:

1. Define RACF groups. Keep in mind the difference between administrative and operational functions.
2. Define terminal operators to RACF.
3. Evaluate the requirement to update the RACF class descriptor table. Any change to it will require an IPL of the MVS system to enable the change.
4. Use the RACF SETROPTS command to activate the CICS/VS classes.
5. Define all non-sensitive DL/I PSBs to RACF. Assign a UACC of READ.
6. Define protected transaction codes to RACF and build their access lists.

All of the above steps can be done before doing anything to CICS/VS.

In CICS/VS:

1. Generate CICS/VS with:

```
DFHSG PROGRAM=CSS

DFHSIT TYPE=INITIAL
    ,EXTSEC=YES
    ,XSP=YES
    ,XTRAN=ttt   (or take the default, CICSTRN)
    ,XPSB=ppp    (or take the default, CICSPSB)
    ,APPLID=     (suggest using userid from JOB statement)
```

2. Generate the CICS/VS sign on table:

```
DFHSNT TYPE=entry
    ,OPNAME=racfid  (7 characters or less)
    ,EXTSEC=YES
```

   For last (or only) entry):

```
DFHSNT TYPE=(entry,default)
    ,EXTSEC=YES
```

3. Generate the CICS/VS transaction specification:

```
DFHPCT TYPE=entry
    ,TRANSID=tttt
    ,EXTSEC=YES
```

4. Initialize CICS/VS with USER=,PASSWORD= in the JOB statement.

5. If access to the CICS/VS region is to be controlled:

```
SETROPTS CLASSACT(APPL)
```

   Define the CICS/VS region's APPLID to RACF and permit appropriate groups to access it.

6. If access to the CICS/VS terminals is to be controlled:

```
SETROPTS CLASSACT(TERMINAL)
         TERMINAL(READ|NONE)
```

   Define the terminals and their access lists to RACF.

7. If access to undefined terminals by CICS/VS operators is not to be allowed:

```
ALTGROUP groupname NOTERMUACC
```

Steps 5 through 7 should be done while the CICS/VS system is inactive; otherwise, operators could be denied access if they sign on between the time the resource is defined and the time the access list is built.

Appendix A. RACF Command Summary and Command Examples

This appendix includes a RACF command summary, a summary of authorities required to issue RACF commands, a summary of installation exits invoked by RACF commands, and a number of examples of command sequences and their effects. While intended primarily for the security administrator, this material should prove useful to group administrators and to the technical support personnel responsible for RACF. In addition, certain portions might prove useful to the end user.

Command Summary and Authority Required to Issue RACF Commands

RACF commands allow you to list, modify, add, and delete profiles for users, groups, connect entries, and resources. Figure A-1 shows, in alphabetic order, each of the commands and its functions. Figure A-2 shows the authorities required to issue commands. See *RACF Command Language Reference* for a complete description of each command and the user requirements.

RACF Command	Command Functions
ADDGROUP	– Define one or more new groups as a subgroup of an existing group. – Specify a model data set profile for a group.
ADDSD*	– Create one or more generic data set profiles. – Create one or more discrete data set profiles and optionally RACF-indicate the DASD data set(s) that the profiles apply to. (Default is to RACF-indicate.) – Create a new data set model profile.
ADDUSER	– Define one or more new users and connect the users to their default connect group. – Specify a model data set profile for a user.
ALTDSD*	– Change one or more discrete or generic data set 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.
ALTGROUP	– Change the superior group of one or more groups. – Change the owner of one or more groups. – Change the terminal indicator for one or more groups.
ALTUSER	– Change one or more users' attributes. – Change one or more users' default universal access authority or level of group authority within a specific group. – Revoke or reestablish one or more users' privileges to access the system. – Change the installation-defined data associated with one or more users. – Alter a model profile name for a user.
CONNECT	– Connect one or more users to a group. – Modify one or more users' connection to a group. – Establish user's authority to modify profiles.
DELDSD*	– Delete a generic data set profile. – Delete a discrete data set profile and optionally remove RACF-indication from the data set the profile applies to. (The default is to remove RACF-indication.)

Figure A-1 (Part 1 of 2). Functions of RACF Commands

RACF Command	Command Functions
DELGROUP*	– Delete one or more groups and their relationship to the superior group.
DELUSER*	– Delete one or more users and remove all their connections to RACF groups.
LISTDSD*	– List the details of one or more discrete or generic data set profiles including the users and groups authorized to access the data set(s) that the profiles apply to.
LISTGRP	– List the details of one or more group profiles including the users connected to the group.
LISTUSER	– List the details of one or more user profiles including all groups each user is connected to.
PASSWORD	– Change a user's password. – Change a user's password change interval. – Reset another user's password to a known default value.
PERMIT* RALTER	– Give authority to access a resource to specific users or groups. – Remove the authority of specific users or groups to access a RACF-protected resource. – 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. – Change the discrete and/or generic profiles for one or more resources whose class is defined in the class descriptor table. – Maintain global tables.
RDEFINE	– create a discrete or generic general resource profile for one or more resources whose classes are defined in the class descriptor table. – Maintain global tables.
RDELETE	– Delete discrete or generic profiles for one or more resources whose classes are defined in the class descriptor table. – Maintain global tables.
REMOVE*	– Remove a user from a group and assign a new owner for any group data sets owned by the user.
RLIST	– List the details of discrete or generic profiles for one or more resources whose class is defined in the class descriptor table. – Maintain global tables.
RVARY	– Dynamically deactivate and reactivate the RACF function. – Deactivate tape volume protection while RACF is deactivated. – Switch the primary and back-up RACF data sets.
SEARCH*	– List the resource names that meet a search criterion for a class of resources. – Create a CLIST using resource names for a class that meets the search criteria.
SETROPTS	– Dynamically set system-wide options relating to resource protection, generic profile checking, terminal universal access authority, statistics gathering, logging of RACF events, and user password expiration interval. – Control global access checking for selected individual resources and/or generic names with selected generalized access rules. – Enable or disable single-level data set name support. – Establish password syntax rules. – Activate password processing for rejecting reused passwords, limiting invalid password attempts, and warning of password expiration. – Activate profile modeling for GDG, group, and user data sets. – Enable or disable list-of-groups processing. – Control the use of automatic data set protection (ADSP). – Display current options in effect. – Initiate refreshing of in-storage generic profiles and global access checking tables.
*Installation exit provided.	

Figure A-1 (Part 2 of 2). Functions of RACF Commands

RACF Command	Required Authority 1	Owner of:			Group Related Authorities 2							User Authority 3				Access Authority			
		User	Group	Resource 5	JOIN	CONNECT	CREATE	USE	SPECIAL 12	OPERATIONS	AUDITOR 6	SPECIAL 4	OPERATIONS	AUDITOR 6	CLAUTH	ALTER 13	CONTROL	UPDATE	READ
ADDSD	7				X	X	X		8	9		X	9						
ADDGROUP	10		X		X				X			X							
ADDUSER	11		X		X				X			X			X				
ALTDSD				X					X		X	X		X		X			
ALTGROUP			X		X				X			X							
ALTUSER	14	X	X		X	X			X		X	X		X					
CONNECT			X		X	X			X			X							
DELDSD				X					X			X				X			
DELGROUP			X		X				X			X							
DELUSER		X							X			X							
LISTDSD				X					X		X	X		X		X	X	X	X
LISTGRP			X		X	X			X		X	X		X					
LISTUSER	15	X							X		X	X		X					
PASSWORD	16	X							X			X							
PERMIT				X					X			X				X			
RALTER				X					X		X	X		X		X			
RDEFINE												X			X				
RDELETE				X					X			X				X			
REMOVE				X	X	X			X			X							
RLIST			X						X			X		X		X	X	X	X
RVARY	17																		
SEARCH				X					X	X	X	X	X	X		X	X	X	X
SETROPTS									18	18	18	X	18	X	18				

Notes:

1. To issue a RACF command the user must be defined to RACF and have sufficient authority as shown in the body of this table. In some cases, additional authority is required to use some command options. See the RACF Command Language Reference for full details.

2. Group authorities are assigned via the CONNECT command. They are only effective when dealing with data sets and profiles related to the group in which the user was assigned the authority.

   JOIN, CONNECT, CREATE and USE attributes are only effective in the group for which they were assigned.

   Group-SPECIAL, group-OPERATIONS, and group-AUDITOR attributes are effective in the group in which they were assigned, all sub-groups of the group, sub-groups of sub-groups and so on. They percolate down through the group tree structure.

3. User authorities are assigned via the ADDUSER and ALTUSER commands and give the user system wide authority. They are not restricted to use on specific profiles.

Figure A-2 (Part 1 of 2). Authorities Required to Issue RACF Commands

4. A user with the SPECIAL attribute is authorized to issue all commands for all profiles, with the exception of those options that are restricted to users with the AUDITOR attribute.

5. For data sets, a user is also considered to be the owner of a profile if the high level qualifier of the profile name is the user's userid. If the owner of the profile is a group, then a user with group-SPECIAL or group-AUDITOR attribute in the group is authorized to modify the profile.

6. A user with the AUDITOR or group-AUDITOR attribute is restricted to setting auditing (logging) options and to listing functions.

7. A user can create data set profiles if the high level qualifier is the user's own userid or a group name in which the user has at least CREATE authority.

8. A group-SPECIAL user can create data set profiles if the high level qualifier is a group in which the user is group-SPECIAL or a userid owned by a group in which the user is group-SPECIAL.

9. An OPERATIONS user can create data set profiles if the high level qualifier is a group name that the user is not connected to with less than CREATE authority.

   A group-OPERATIONS user can create data set profiles if the high level qualifier is a group in which the user is group-OPERATIONS.

10. The user must be the owner of the superior group, or be connected to the superior group with JOIN or group-SPECIAL authority.

11. The user must have class authority in the USER class and (a) be the owner of the default group, or (b) be connected to the default group with JOIN or group-SPECIAL authority.

12. A user with the group-SPECIAL, group-OPERATIONS or group-AUDITOR attributes in a group is authorized to the profile for:

    - The group,
    - Subgroups of the group,
    - Users owned by the group,
    - Data sets owned by the group,
    - Data sets with a high level qualifier that is the group name,
    - Data sets owned by users who are owned by the group,
    - Data sets with a high level qualifier that is a userid owned by the group,
    - General resources owned by the group,
    - General resources owned by users who are owned by the group.

    Group authority automatically percolates to sub-groups of the group all the way down the group tree structure.

13. ALTER authority is normally required to modify a profile and to list it's access list. For generic profiles, the user must be authorized by ownership or user or group authorities.

14. All users can modify their own name fields and default groups.

15. All users can list their own user profiles.

16. All users can set their own password and password interval.

17. No authority is required for this command. However the operator (at the master console or security console) must approve the request.

18. System options can not be changed except by users with the SPECIAL or AUDITOR attributes.

Figure A-2 (Part 2 of 2). Authorities Required to Issue RACF Commands

Installation Exits Invoked by RACF Commands

Four preprocessing installation exits are invoked by certain RACF commands as shown in the following table.

	Exit Name	Invoked By	When Entered (after syntax checking and ...)
Command exit	ICHCNX00	ADDSD command	before any authorization checking is performed
		ALTDSD command	before the DATASET profile is retrieved
		DELDSD command	before the DATASET profile is retrieved
		LISTDSD command	before any DATASET profile is located for the ID, PREFIX, or DATASET parameters, and after each DATASET profile is retrieved but before any authorization checking is performed
		PERMIT command	before the DATASET profile is retrieved
		SEARCH command	before the first DATASET profile is retrieved, and after each DATASET profile is located but before any authorization checking is performed
		ICHUTI00 utility	after the DATASET profile is retrieved, but before the DATASET profile is associated with a user or group.
Report writer exit	ICHRSHFE	RACF report writer	after the SMF record is read, but before the data set name or generic profile name is compared to selection criteria
Command exit	ICHCCX00	DELGROUP command	before any DATASET profile is located
		DELUSER command	before any DATASET profile is located
		REMOVE command	before any DATASET profile is located
Command exit	ICHPWX01	PASSWORD command	during password validation
		ALTUSER command	during password validation

You can use the ICHCNX00 installation exit routine to maintain a DASD data set naming convention other than that used by RACF. The routine can:

- Supply a value to be used as the high-level qualifier for authorization checking. For example, you could extract that userid from the second-level qualifier of the data set name to use as the qualifier for authorization checking.
- Bypass any authorization checking of the high-level qualifier.
- Perform additional security checking.
- Bypass all security checking.

You can use the ICHCCX00 installation exit routine to control cleanup procedures when using other than RACF DASD data set naming conventions. The exit routine can ensure that a user or group being deleted does not own data sets defined to RACF and that a user being removed from a group does not own any group data sets. The routine can:

- Modify the data set search
- Bypass any data set search
- Fail the request

You can use the ICHPWX01 installation exit routine to examine the intended new password and the new password change interval (if invoked from the PASSWORD command). In the case of new password processing by the password command, the exit routine gains control if all the following conditions are true.

- The new password is not a duplicate of a current password.
- The new password is not a duplicate of a previous password (if the password history option is active).
- The new password conforms to all installation syntax rules.

Table-Driven Data Set Naming Conventions

RACF offers installations the ability to create a naming convention table. You can use the naming convention table to set up and enforce a data set naming convention other than that used by RACF. The table can:

- Supply a qualifier to be used as the high-level qualifier for authorization checking
- Convert data set names to RACF naming convention form for RACF use
- Convert names in RACF form to the installation's format for external display
- Enforce a naming convention by not allowing the definition of data sets that do not conform to an installation's rules
- Reduce RACF overhead by determining whether a data set is a user or group data set

Naming convention processing is done by the ICHNRT00 routine immediately before the current preprocessing/naming convention installation exits are called. (The exits can still be used for additional processing.) ICHNRT00 uses a table that is created by the assembler language macro (ICHNCONV). See *SPL: RACF* for a full description of its use.

The convention processing routine, ICHNRT00, is called before the following installation exits are called:

- ICHRDX01 - RACDEF preprocessing exit
- ICHRCX01 - RACHECK preprocessing exit

- ICHCNX00 - Command naming convention exit in:
  - ADDSD
  - ALTDSD
  - DELDSD
  - LISTDSD
  - PERMIT
  - SEARCH
  - ICHUT100

The RACF table-driven naming convention feature largely replaces the need for the ICHCNX00 exit routine. (The naming convention table is processed before each call to ICHCNX00.) See *SPL: RACF* for a full description of the naming convention table.

RACF Command Examples

The easiest way to learn how to use the RACF commands is to use them interactively. This section shows some common command sequences and their effects; it is intended for use by security administrators, group administrators, and system programmers responsible for RACF. Certain sequences can be useful for end users. Each sequence is heavily commented to show rationale and the assumptions made about previous commands. Comments and commands are interspersed. Commands appear on a separate line and are preceded by an indicator string of **-->**. The sequences are:

- Initial RACF command session
- Defining users and groups
- Protecting system data sets
- Protecting group data sets
- Protecting user data sets
- Controlling auditing
- Using DASDVOL protection
- Deleting users
- Deleting groups
- Using the SEARCH command

Note: The sample sessions assume that all the users defined in the sessions are already authorized to log on to TSO; that is, they are defined in the UADS.

Session 1: Initial RACF Command Sequences

When the RACF data set is first initialized, it has the following groups defined:

Group	Superior Group/Owner		Connected Users (Authority)
SYS1	--	IBMUSER	IBMUSER (JOIN)
VSAMDSET	SYS1	IBMUSER	IBMUSER (JOIN)
SYSCTLG	SYS1	IBMUSER	IBMUSER (JOIN)

And there is only one user.

User	Default Group	Attributes/Connected Groups (Authority)
IBMUSER	SYS1 (JOIN)	SPECIAL,OPERATIONS SYSCTLG (JOIN) VSAMDSET (JOIN)

IBMUSER's temporary password is equal to the default group (SYS1). In the first session, you change the password. You, the RACF security administrator, log on as IBMUSER:

```
--> logon ibmuser
```

After you enter your old password (sys1) and a new password, you define a new user (RACFADM) to RACF for your own use, with the SPECIAL, OPERATIONS, and AUDITOR user attributes:

```
--> adduser racfadm special operations auditor
```

You then log on as that user, entering your old password (sys1) and a new password:

```
--> logon racfadm
```

You first list all users to ensure that only RACFADM and IBMUSER are defined to RACF, and that they have the proper attributes.

```
--> listuser *
```

You then list all the groups defined to RACF:

```
--> listgrp *
```

Connect yourself to them and make yourself the owner of the groups:

```
--> connect racfadm group(sysctlg) auth(join)
--> connect racfadm group(vsamdset) auth(join)
--> altgroup vsamdset owner(racfadm)
--> altgroup sysctlg owner(racfadm)
--> altgroup sys1 owner(racfadm)
```

You then revoke IBMUSER so that another user cannot make use of the IBMUSER userid:

```
--> altuser ibmuser revoke
```

You define a user to RACF (userid 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
```

You proceed to add four groups: three (GROUP1, GROUP2, and GROUP3) will be departmental groups, with GROUP2 and GROUP3 owned by GROUP1 so that certain authorities can be propagated. The fourth group (DATAMGT) will have global pack maintenance responsibility.

```
--> addgroup (group1 datamgt)
--> addgroup (group2 group3) owner(group1) sup(group1)
```

You now define a global RACF auditor who will have system-wide auditing responsibilities and privileges.

```
--> adduser audccc auditor
```

You now add a user (D03DlK) to GROUP3 with authority to protect resources.

```
--> adduser do3dlk owner(group3) auth(create) dfltgrp(group3)
```

You proceed to define a group administrator with the group-SPECIAL attribute for each group. Only the administrator for GROUP1 will have authority to define new users. Each of the other administrators will have authority over resources owned by his group, as well as resources owned by users who are owned by his 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) auth(create) special
--> adduser d03abl dfltgrp(group3) data('group3 adm')
--> connect d03abl group(group3) auth(create) special
```

You proceed to define a user who will have auditor privileges in GROUP1, GROUP2, and GROUP3. You connect the user to GROUP1 and give the user the group-AUDITOR attribute. Because GROUP2 and GROUP3 are owned by GROUP1, the user will have auditor authority over the resources and users belonging to those groups, as well as to GROUP1. The user will 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, will be able to define DASD volumes to RACF in order to perform dump, restore, and pack cleanup operations.

```
--> adduser dmgjfs dfltgrp(datamgt) auth(join) +
-->     clauth(user dasdvol) data('data mgt adm')
```

Because of his duties, the data manager is connected to SYS1, allowing the manager to access data sets with SYS1 in their access list and to define SYS1 data set profiles to RACF. When logged on to SYS1, the data manager will have the group-SPECIAL attribute.

```
--> connect dmgjfs group(sys1) auth(create) uacc(read) +
-->     special
```

At the end of the session, the defined group structure is:

Group	Superior Group	Owner	Connected Users	(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	
GROUP2	SYS1	GROUP1	D02JMP	
GROUP3	SYS1	GROUP2	D03ABL	
DATAMGT	SYS1	RACFADM	DMGJFS	(JOIN)

The defined users are:

User	Default Group		Attributes/Connected Groups (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)	CLAUTH(USER DASDVOL)
	SYS1	(CREATE)	SPECIAL
D01RHG	GROUP1	(JOIN)	CLAUTH(USER), group-SPECIAL, (JOIN)
D02JMP	GROUP2		group-SPECIAL
D03ABL	GROUP3		group-SPECIAL
D01GPB	GROUP1		group-AUDITOR
AUDCCC	SYS1	USE	AUDITOR

Session 2: Defining Users and Groups

The group administrator for GROUP1 (userid d01rhg, old password group1) logs on to define a subgroup for a project and to define new users.

```
--> logon d01rhg
```

The group administrator defines a subgroup,

```
--> addgroup group11 supgroup(group1) owner(group1)
```

creates a model profile for group11,

```
--> addsd 'group11.model' model owner(group11)
```

and defines a user naming the new group as the default and owning group.

```
--> adduser d01fxk dfltgrp(group11) auth(join) +
--> name(f.x.kearny) owner(group11)
```

The group administrator then defines three users to RACF. Their default group and owning group will be GROUP2.

```
--> adduser d02sal dfltgrp(group2) auth(create) +
--> name(s.a.lory) owner(group 2)
--> adduser d02lxs dfltgrp(group2) auth(create) +
--> name(l.smith) owner(group2)
--> adduser d02spk dfltgrp(group2) auth(create) +
--> (s.p.kennedy) owner(group2)
```

Session 3: Protecting System Data Sets

The data manager logs on to TSO within the data management group (old password datamgt).

```
--> logon dmgjfs
```

The data manager starts by protecting the RACF data set with a discrete RACF profile,

```
--> addsd 'sys1.racf' uacc(none)
```

and listing the RACF description of the data set to look at the access list.

```
--> listdsd da('sys1.racf') auth
```

The data manager then decides, because several system data sets will have the same access list, to build a model access list and copy it as needed. The manager builds a model profile,

```
--> addsd 'sys1.data.model' model
```

and builds an access list for the model profile. (Note: It is also possible to make use of the model profile automatically by specifying the model name on an ALTGROUP command.)

```
--> permit 'sys1.data.model' id(sys1 datamgt) access(alter)
```

The data manager then protects some system data sets with discrete profiles,

```
--> addsd 'sys1.haspace' uacc(none)
--> addsd 'sys1.uads' uacc(none)
--> addsd 'sys1.jobs' uacc(none)
```

and builds their access lists, referring to the model profile.

```
--> permit 'sys1.haspace' from('sys1.data.model')
--> permit 'sys1.uads' from('sys1.data.model')
--> permit 'sys1.jobs' from('sys1.data.model')
```

The data manager then protects the broadcast data set so that it can be updated by all users,

```
--> addsd 'sys1.brodcast' uacc(update)
```

and then protects the rest of the system data sets with a generic profile so that they may be read by any user but allocated or deleted by only the SYS1 group.

```
--> addsd 'sys1.*' uacc(read)
--> permit 'sys1.*' id(sys1) access(alter)
```

To allow most system data sets to be read without reference to the RACF data set, the system data sets are also defined for global access checking.

Note that each data set having universal access authority other than READ must have a specific entry defined to the global access checking facility. All other data sets with a high level qualifier of SYS1 can be covered with a single generic global access checking entry.

```
--> rdefine global dataset addmem('sys1.racf'/none)
--> ralter global dataset addmem('sys1.uads'/none)
--> ralter global dataset addmem('sys1.brodcast'/update)
--> ralter global dataset addmem('sys1.haspace'/none)
--> ralter global dataset addmem('sys1.jobs'/none)
--> ralter global dataset addmem('sys1.*'/read)
```

Note: See Chapter 4, "Protection Via Generic Profiles," for information on coverage and restrictions.

Session 4: Protecting Group Data Sets

The group administrator for GROUP1 logs on to protect the group's data sets.

```
--> logon d01rhg
```

After entering the passwords, the group administrator starts by protecting the group's master file data set with a discrete RACF profile,

```
--> addsd 'group1.database' uacc(none)
```

and listing the RACF description of the data set to look at the access list.

```
--> listdsd da('group1.database') auth
```

The group administrator then decides, because several group data sets will have the same access list, to build a model access list and copy it as needed. The administrator builds a model profile,

```
--> addsd 'group1.grp1data.model' model
```

builds an access list for the model profile,

```
--> permit 'group1.grp1data.model' id(sys1 datamgt) access(alter)
--> permit 'group1.grp1data.model' id(group1) access(update)
```

and lists the access list of the model to make sure it is correct.

```
--> listdsd da('group1.grp1data.model') auth
```

The administrator then copies the model access list to the access list of the group's master file data set,

```
--> permit 'group1.database' from(group1.grp1data.model)
```

and lists the access list to make sure it is correct.

```
--> listdsd da('group1.database') auth
```

The administrator then defines some other group1 data sets to RACF. These data sets require discrete profiles for statistical purposes.

```
--> addsd 'group1.datax' uacc(none) audit(failures(alter))
--> addsd 'group1.datay' uacc(none) audit(success(update)failures(re
--> addsd 'group1.dataz' uacc(none) audit(all)
```

Note that their access list is automatically copied from the group's model, GROUP1.GRP1DATA.MODEL.

The administrator then protects the rest of the GROUP1 data sets with generic profiles so that they can be accessed at the ALTER authority level by all members of GROUP1, but not accessed at all by other users. An exception will be those data sets having at least three qualifiers starting with GROUP1.LOW; these data sets must be able to be updated by GROUP 3 and read by any other users.

First the administrator deletes the model profile,

```
--> deldsd 'group1.grp1data.model'
```

then adds the two profiles,

```
--> addsd 'group1.*' uacc(none)
--> addsd 'group1.month%%.*' uacc(read)
```

then builds the access lists, deleting the existing access list produced by modelling using the RESET option.

```
--> permit 'group1.*' id(group1) access(alter) reset
--> permit 'group1.month%%.*' id(group3) access(update) reset
```

The administrator then realizes that a certain subset of the group's data sets (those having names of the structure "GROUP1.XYX.etc") will need a more specific access list than that provided by the "GROUP1.*" profile. However, the administrator does not know for sure which of the users access these data sets in the course of their normal work activities. The administrator decides to log accesses to those data sets for a time, while allowing full access by all members of his group.

```
--> addsd 'group1.xyx.*' uacc(alter) audit(all)
```

After a time of reviewing accesses to these data sets through reports supplied by the RACF system auditor, the administrator decides to build the access list, but to initially issue warning messages instead of access failures in order to "test"the access list.

```
--> altdsd 'group1.xyx.*' warn
--> permit 'group1.xyx.*' id(d02jmp) access(read)
```

After the administrator has "tested"the access list and is satisfied that it is correct, he will use the altdsd command to remove the warn option from the profile and actually protect the data.

Note: See Chapter 4, "Protection Via Generic Profiles," for information on coverage and restrictions.

Session 5: Protecting User Data Sets

A user logs on within GROUP3 (old password group3).

```
--> logon d03dlk
```

The user decides to protect all of his data sets from all other users.

```
--> addsd 'd03dlk.*' uacc(none)
```

In addition to protecting all existing data sets with the high level qualifier of 'd03dlk', the generic profile automatically protects new data sets as they are allocated. The generic profile also restricts use of the qualifier so that other users cannot allocate new data sets with it or rename existing data sets to it.

The user then decides to allow the rest of his group to read but not modify his data sets. The user puts the rest of his group in the access list with READ authority.

```
--> permit 'd03dlk.*' id(group3) access(read)
```

The user lists the generic profile and access list.

```
--> listdsd dataset('d03dlk.*') all
```

The user then decides that an exception to the imposed security is necessary; GROUP2 users must be able to scratch, allocate, and access the subset of his data sets whose names begin with 'd03dlk.group2'. GROUP2 users must not be able to access any of his other data sets. The user defines the generic profile

```
--> addsd 'd03dlk.group2.*' uacc(none)
```

and allows his group to read these data sets.

```
--> permit 'd03dlk.group2.*' id(group3) access(read)
```

The user allows GROUP2 to have full control over the use and disposition of these data sets.

```
--> permit 'd03dlk.group2.*' id(group2) access(alter)
```

The user lists the generic profile and access list.

```
--> listdsd da('d03dlk.group2.*') all
```

Note: See Chapter 4, "Protection Via Generic Profiles," for information on coverage and restrictions.

Session 6: Controlling Auditing

The auditor (userid AUDCCC) logs on to set logging options.

```
--> logon audccc
```

The auditor decides to log changes to user and group definitions, to log activities of users with the SPECIAL attribute, and to log RACF command violations.

```
--> setropts audit(user group) saudit cmdviol
```

The auditor also decides to log the RACF actions of group administrators.

```
--> altuser dmgjfs uaudit
--> altuser d01rhg uaudit
--> altuser d02jmp uaudit
--> altuser d03abl uaudit
```

The auditor decides to ensure that, at a minimum, failures to access data sets will be logged. The auditor uses the SEARCH command to construct a CLIST, setting the GLOBALAUDIT option for all the data sets defined to RACF and executes the CLIST.

```
--> search nomask nolist +
-->     clist('altdsd ' ' globalaudit(failures(read))')
--> exec exec.racf.clist
```

Session 7: Using DASDVOL Authorization

The data manager logs on to test DASDVOL authorization.

```
--> logon dmgjfs
```

The manager first determines if DASDVOL authorization is active.

```
--> setropts list
```

The command fails because the data manager is not authorized to issue the command. The data manager allocates a data set on a specific volume

```
--> allocate 'sys1.x.data' vol(111111) sp(1) new track
```

and protects the data set with a discrete profile.

```
--> addsd da('sys1.x.data') uacc(none)
```

The data manager lists the RACF profile for the data set to display the access list.

```
--> listdsd da('sys1.x.data') auth
```

The data manager is in the access list with ALTER authority, because he defined the data set. He uses PERMIT to take himself out of the access list,

```
--> permit 'sys1.x.data' id(dmgjfs) delete
```

and lists the data set profile again to ensure that the access list is empty.

```
--> listdsd da('sys1.x.data') auth
```

The data manager is still the owner of the data set and can execute RACF commands but cannot open or delete the data set.

```
--> delete ('sys1.x.data')
```

The manager then defines to RACF the volume on which the data set resides

```
--> rdefine dasdvol 111111 uacc(none)
```

and lists its description.

```
--> rlist dasdvol 111111 auth
```

Because the data manager defined the DASD volume, he is in the access list with ALTER. The manager again tries to delete the profile for the data set, but because DASDVOL checking is not in effect, the delete fails again.

```
--> delete ('sys1.x.data')
```

Session 8: Deleting Users

The RACF security administrator (userid RACFADM) logs on to delete a user. The user has already been revoked from the system by the user's group administrator.

```
--> logon racfadm
--> deluser d02sal
```

If the command fails, the user still owns data sets. That is, there are data sets defined to RACF with the user's userid (D02SAL) as a high-level qualifier. You decide not to scratch the data sets but to remove their RACF protection and use the SEARCH command to construct an appropriate TSO CLIST.

```
--> search mask(d02sal) clist('deldsd')
```

You list the CLIST

```
--> l exec.racf.clist
```

(note that "I" means "list") and then execute it.

```
--> exec exec.racf.clist
```

This time you can delete the user.

```
--> deluser d02sal
```

Session 9: Deleting Groups

The GROUP1 administrator (userid D01RHG) logs on

```
--> logon d01rhg group(group1)
```

and tries to delete the subgroup GROUP11.

```
--> delgroup group11
```

The delete request fails, because the group owns users or resource profiles. The administrator lists the group definition.

```
--> listgrp group11
```

The group definition includes the user d01fxk who is owned by the group. This user must be deleted.

```
--> del user d01fxk group(group11)
```

The administrator again tries to delete the group,

```
--> delgroup group11
```

but this time there are group data set profiles that exist (that is, data set profiles defined to RACF that have a high-level qualifier equal to the group name, GROUP3). The administrator lists their RACF description, using the PREFIX operand of the LISTDSD command.

```
--> listdsd prefix(group11)
```

Only one data set profile exists and the administrator deletes its description,

```
--> deldsd 'group11.model'
```

and again tries to delete the group.

```
--> delgroup group11
```

This time the attempt succeeds.

Session 10: SEARCH Command

The RACF SPECIAL user logs on to find the userids that have not accessed the system in the last 90 days.

```
--> logon racfadm
```

The first job is to create a list of userids.

```
--> search class(user) clist age(90)
```

The list of the userids (if any) has been placed in the data set RACFADM.EXEC.RACF.CLIST. This data set can then be used for administrative cleanup. To obtain more information about the userids, the SPECIAL user enters the following commands.

```
--> search class(user) clist('listuser ') age(90)
--> exec exec.racf
```

This produces a complete listing about each userid that has not accessed the system in the last 90 days.

Index

C

D

M

Q

R

INDE

T

V

W

X

SC28-1340-1

IBM®

Resource Access Control Facility (RACF)
Security Administrator's Guide

SC28-1340-1

READER'S COMMENT FORM

This manual is part of a library that serves as a reference source for system analysts, programmers, and operators of IBM systems. 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.***

Possible topics for comment are:

Clarity Accuracy Completeness Organization Coding Retrieval Legibility

If you wish a reply, give your name, company, mailing address, and date:

_____

_____

_____

_____

Note: Staples can cause problems with automated mail sorting equipment. Please use pressure sensitive or other gummed tape to seal this form.

Cut or Fold Along Line

What is your occupation? _____

How do you use this publication? _____

Number of latest Newsletter associated with this publication: _____

Thank you for your cooperation. No postage stamp necessary if mailed in the U.S.A. (Elsewhere, an IBM office or representative will be happy to forward your comments or you may mail directly to the address in the Edition Notice on the back of the title page.)

Reader's Comment Form

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 PERMIT 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE:

International Business Machines Corporation
Department D58, Building 920-2
PO Box 390
Poughkeepsie, New York 12602

Fold and tape | Please Do Not Staple | Fold and tape

IBM®

...Security Administrator's Guide (File No. S370-34) Printed in U.S.A. SC28-1340-1