OS/390

# SecureWay Security Server RACF Callable Services

GC28-1921-08

OS/390

# SecureWay Security Server RACF Callable Services

GC28-1921-08

**Note**
Before using this information and the product it supports, be sure to read the general information under "Appendix. Notices" on page 181.

**Ninth Edition, December 2000**

This is a major revision of SC28-1921-07.

This edition applies to Version 2 Release 10 of OS/390 (5647-A01) and to all subsequent releases and modifications until otherwise indicated in new editions.

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

IBM welcomes your comments. A form for readers' comments may be provided at the back of this publication, or you may address your comments to the following address:

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

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

IBMLink (United States customers only): IBMUSM10(MHVRCFS)
Internet e-mail: mhvrcfs@us.ibm.com
World Wide Web: http://www.ibm.com/s390/os390/webqs.html

If you would like a reply, be sure to include your name, address, telephone number, or FAX number.

Make sure to include the following in your comment or note:
- Title and order number of this book
- Page number or topic related to your comment

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

# Contents

# Tables

# About This Book

This book contains information about the Resource Access Control Facility (RACF), which is part of theSecureWay Security Server for OS/390 The Security Server has these components:

- RACF
- DCE Security Server
- OS/390 Firewall Technologies
- Lightweight Directory Access Protocol (LDAP) Server, which includes client and server function
- Open Cryptographic Enhanced Plug-ins (OCEP)
- SecureWay Security Server Network Authentication and Privacy Service

For information about these components, see the publications related to them.

This book documents callable services provided by RACF. It contains:

- Information on using the callable services
- A description of each callable service
- Descriptions of the data areas used by the callable services
- A description of an installation exit that can be used in conjunction with the callable services

## Who Should Use This Book

This book is intended for system programmers who are familiar with RACF concepts and terminology. They should also be familiar with MVS systems and with OS/390 UNIX System Services.

# Summary of Changes

**Summary of Changes**
**for GC28-1921-08**
**OS/390 Version 2 Release 10**
**As Updated December, 2000**

This book contains information previously presented in *OS/390 Security Server (RACF) Callable Services*, GC28-1921-07.

The following summarizes the changes to that information.

**New Information**

- The new callable service, R_PKIServ (IRRSPX00) has been added.

This book includes terminology, maintenance, and editorial changes. Technical changes or additions to the text and illustrations are indicated by a vertical line to the left of the change.

**Summary of Changes**
**for GC28-1921-07**
**OS/390 Version 2 Release 10**

This book contains information previously presented in the *OS/390 Security Server (RACF) Callable Services*, GC28-1921-06, which supports OS/390 Version 2 Release 8.

The following changes appear only in the online version of this publication.

**New Information**

- New parameters, X500name and Variable_list, have been added to callable service, initACEE (IRRSIA00). Information has also been added about using certificate mapping profiles with the APPL_id parameter.
- The new callable service, R_kerbinfo (IRRSMK00) has been added.
- The new callable service, R_ticketserv (IRRSPK00) has been added.

**Changed Information**

- Callable service, R_admin (IRRSEQ00), is changed to support the restricted access attribute in the user base segment. FieldName and REST have been added.
- Callable service, initACEE (IRRSIA00), now includes information for X500name in the RACF Authorization Usage Notes section.
- Callable service, R_admin (IRRSEQ00), is changed to support new ADDUSER/ALTUSER and RDEFINE/RALTER keywords.
- Callable service, R_usermap (IRRSIM00), is changed to allow the specification of SecureWay Security Server Network Authentication and Privacy Service identity information and will map a SecureWay Security Server Network Authentication and Privacy Service principal name to a RACF userid and a RACF userid to SecureWay Security Server Network Authentication and Privacy Service.
- Callable service, ck_priv (IRRSKP00), is changed to support new CHMOUNT function codes.
- Callable service, R_admin (IRRSEQ00), has been updated to include new RACF Authorization information, Return and Reason Codes, and new Usage Notes.

- Callable service, R-Datalib (IRRSDL00), has been updated to include new RACF Authoriztion information, new function code for Parameters, new Return and Reason Codes, and updated Usage Notes.
- Callable service, initACEE (IRRSIA00), has been updated to include new and changed RACF Authorization information, and new and changed Usage Notes.
- Callable service, R_fork (IRRSFK00), has been updated to include a new reason code and indicates there is additional security information associated with the parent process.

This book includes terminology, maintenance, and editorial changes, including the following:

The OS/390 Security Server, of which RACF is a component, has joined the IBM SecureWay family of products. As such, occurrences of OS/390 Security Server have been changed to SecureWay Security Server for OS/390, or its abbreviated name, Security Server. OS/390 Security Server may continue to appear in messages, panel text, and other code with SecureWay Security Server for OS/390.

Technical changes or additions to the text are indicated by a vertical line to the left of the change.

**Summary of Changes**
**for GC28-1921-06**
**OS/390 Version 2 Release 8**

This book contains information previously presented in the *OS/390 Security Server (RACF) Callable Services*, GC28-1921-05, which supports OS/390 Version 2 Release 6 and OS390 Version 2 Release 7.

**New Information**

The new callable service, R_datalib (IRRSDL00) has been added.

The new callable service, R_usermap (IRRSIM00) has been added.

Running headers provided for each callable service.

The following new Index entries have been added:
_POSIX_CHOWN_RESTRICTED, _POSIX_SAVED_IDS, and NGROUPS_MAX.

**Changed Information**

In the description of clear_setid (IRRSCS00), the note stating that the CRED user type must be local has been removed.

The initACEE (IRRSIA00) callable service has been updated to include new Format information, Parameters, Return Codes, and Usage Notes.

The initUSP (IRRSIU00) callable service has been updated with a new Usage Note, and new Function information.

The makeFSP (IRRSMF00) callable service has been updated with new Usage Notes.

In the description of makeFSP (IRRSMF00), the note stating that the CRED user type must be local has been changed.

The query_file_security_options (IRRSQF00) callable service has been updated with new information.

The R_admin (IRRSEQ00) callable service has been updated with new Parameter List information, new OMVS, LNOTES and NDS Segment Fields, and a new BASE Segment Field entry of RETPD. The PASSWORD field of the Base Segment Fields Table has been updated to include the Flag Byte Value of 'N'. Updated Usage Notes.

The R_admin (IRRSEQ00) callable service has been updated with three new Function Code Values for SETROPTS information, and new tables for the following: Parameter List Mapping for SETROPTS Administration, Segment Entry Fields, Field Entry Format, Base Segment Field Names, and, Output Message Block, as well as supporting detail for these new tables.

In the description of R_chaudit (IRRSCA00), the note stating that the CRED user type must be local has been removed.

The R_chown (IRRSCO00) callable service has new RACF Authorization information for _POSIX_CHOWN_RESTRICTED, and a new Related Services entry of query_file_security_options.

The R_dceruid (IRRSUD00) callable service has a new Related Services entry of R_usermap.

A new section, **RACF Authorization** has been added to every callable service.

The new UNIXPRIV class resource table has been added to the following callable services: R_chown, R_IPC_ctl, R_ptrace, ck_access, ck_owner_two_files, ck_priv, ck_process_owner, and, query_file_security_options.

The IRRSXT00 Installation Exit chapter has been updated to add new Function information, Interface Registers detail, Input information, Output information, and a new Usage Note.

**Summary of Changes**
**for GC28-1921-05**
**OS/390 Version 2 Release 6**

This edition adds three parameters to initACEE to support transportable security environment and OS/390 UNIX multiprocess/multiuser enhancements. For the transportable security environment enhancement, these parameters are ENVR_in and ENVR_out. For OS/390 UNIX multiprocess/multiuser enhancement, it is Output_area. The initACEE callable service also provides an interface for registering and deregistering certificates through the OS/390 UNIX System Services __security service, and for querying a certificate to determine if it is associated with a user ID.

The R_admin callable service has been updated to add support for adding, altering, deleting, and listing data set, general resource, and group profiles. These support enhancements to TME 10.

Changes have also been made to the following callable services to support task level processes: ck_IPC_access, ck_process_owner, makeISP, R_exec, R_fork, R_IPC_ctl, R_ptrace. This supports OS/390 UNIX System Services multiprocess/multiuser

As part of the name change of OpenEdition to OS/390 UNIX System Services, occurrences of OpenEdition have been changed to OS/390 UNIX System Services or its abbreviated name, OS/390 UNIX where appropriate. OpenEdition may continue to appear in messages, panel text, and other code locations.

Each technical change is indicated by a vertical line to the left of the change.

# Chapter 1. Using the RACF Callable Services

The RACF security functions provided for use by OS/390 UNIX System Services and other products integrated with it are called as callable services. Normal installation applications using the services or functions of OS/390 UNIX System Services cannot call the RACF callable services directly. They must use the OS/390 UNIX System Services callable services instead.

## Linkage Conventions for the Callable Services

The linkage is created as follows:

For non-IBM modules, or modules written in languages other than PL/X, the CALL statement must generate a V-type constant (VCON) with the module name of a stub routine for the requested service. The module names are defined as part of the callable services interface described in "Chapter 2. Callable Services Descriptions" on page 9. The VCON can be resolved by link-editing the control section (CSECT) with the stub routines provided as part of MVS's system authorization facility (SAF). There is a stub for each service.

The linkage loads a function code indicating the service requested and calls the callable services router. The function codes that can be used are described in *OS/390 SecureWay Security Server RACF Data Areas*.

The callable services router calls an installation exit (IRRSXT00) and then calls the RACF router.

The RACF router invokes the requested service routine based on the function code.

The service routine provides the requested function and returns to the SAF callable services router.

The SAF callable services router calls the installation exit IRRSXT00 a second time, sets the SAF return code, and returns to the caller.

## Work Area (WORK)

When a module calls a RACF callable service, it must provide the address of a work area. The work area is a 1024-byte structure that is used by SAF, RACF, and the SAF exit routine IRRSXT00. IRRSXT00 can use the first 152 bytes of the area. The first 16 bytes are preserved from the pre-RACF exit invocation to the post-RACF exit invocation and can be used to pass parameters.

For the mapping of the work area, see *OS/390 SecureWay Security Server RACF Data Areas*. For information on IRRSXT00, see "Chapter 3. IRRSXT00 Installation Exit" on page 177.

## File Security Packet (IFSP)

Security-relevant data for files in the OS/390 UNIX System Services file system is kept in a file security packet (IFSP) structure owned by RACF. The IFSP is stored in the file system as part of the attributes associated with a file. When a file is created, the IFSP is created by the **makeFSP** or the **make_root_FSP** callable service.

The **makeFSP** service returns an IFSP to the file system, which writes it with other attributes of the file. On subsequent accesses to the file, the file system reads the IFSP and passes it to other callable services. The file system deletes the IFSP when the file is deleted.

The IFSP is a fixed-size 64-byte area. It is written to storage as part of the PFAR for the file and its size cannot be changed.

The file system manages the storage for the IFSP. The **makeFSP** service fills in the data, and other callable services use or modify the data in the area provided by the file system.

The IFSP data can be examined by users other than the security product. The IFSP is mapped by macro IRRPIFSP. Others should not use this mapping to create or directly modify the IFSP, and should not make their own security or audit decisions based on the contents of the IFSP.

The IFSP contains the following data:

- Control block ID
- Version number
- OS/390 UNIX user identifier (UID) of the owner of the file
- OS/390 UNIX group identifier (GID) of the group owner of the file
- Mode bits:
  - Owner permission bits
  - Group permission bits
  - Other permission bits
  - S_ISUID, S_ISGID, and S_ISVTX bits
- User audit options for the file
- Auditor audit options for the file

For the mapping of the file security packet, see *OS/390 SecureWay Security Server RACF Data Areas*.

## Security Credentials (CRED)

The security credentials (CRED) structure is used in the OS/390 UNIX System Services file system to pass data from the logical file system (LFS) through the physical file system (PFS) to the RACF callable services.

The CRED is built by the LFS, and is created for each system call entry to the LFS. The CRED is used for all vm_ops called (and most RACF callable service calls by the PFS) for the system call. The CRED is not kept across multiple LFS system calls.

The CRED contains:

- **User information:** a user type field that indicates whether the caller is a standard OS/390 UNIX System Services process known to RACF, or a system function that is not a process.

  Functions that accept a system caller process the request as if the caller is a superuser. If an audit record is written, the user OS/390 UNIX user identifier (UID) and OS/390 UNIX group identifier (GID) values in the record are set to -1.

- **Audit data:** data known by the LFS that needs to be passed through the PFS to the RACF callable services for auditing. This data is:
  - **Audit function code:** a code that identifies the system call being processed. The audit function codes are described in *OS/390 SecureWay Security Server RACF Data Areas*.
  - **Name flag:** a flag used on path resolution calls to **ck_access** to indicate whether the first or second file name is being checked.
  - **Requested path name:** the path name the user passed on the system call. For link, vlink, rename, and vrename, this is the old path name. When the caller of lookup is getcwd, ioctl, or ttyname, this field is not filled in.
  - **File name** the part of the requested path name currently being checked. This may be part of the path name or may be part of a symbolic link encountered when resolving the path name. The first directory checked in a path name resolution is either the root directory (**/ROOT**) or the current working directory (**/CWD**). The names **/ROOT** and **/CWD**—the only file names that contain a slash (**/**)—are provided to indicate these directories in the audit record. This field is included only in audit records produced by **ck_access**.

    This field contains the file name of:

    - The directory being checked on calls from lookup.

      When the caller of lookup is getcwd, ioctl, or ttyname, this field is not filled in.
    - The parent directory of the object identified by the pathname for calls for mkdir, mknod, vcreate, open(new file), rename, vrename, rmdir, symlink, vsymlink, unlink, and vremove.
    - The object identified by the path name for calls for open(old file), opendir, link, vlink, and utime.
    - **Second path name:** for rename, vrename, link, and vlink, this is the new path name passed on the system call. For symlink and vsymlink, this is the content of the symlink. For mount and unmount, this is the data set name of the HFS data set being mounted or dismounted.
    - **Second file name:** this is the same as the file name above, except that it is for the second part of the path name being checked. This field contains the file name of:
      - The directory being checked on calls from lookup
      - The parent directory of the object identified by the new pathname for calls for link, vlink, rename, and vrename.

The CRED structure is mapped by the IRRPCRED mapping macro.

For the mapping of the CRED, see *OS/390 SecureWay Security Server RACF Data Areas*.

## File Identifiers

**Programming Interface**

**Programming Interface**

Part of the audit data for file access is a file identifier. The file identifier is a 16-byte token that uniquely identifies a file while it is mounted on the system.

**End of Programming Interface**

**Programming Interface**

If the file system is unmounted and remounted, that file identifier may change. A change in file identifiers can be detected in the audit trail by matching the mount audit records with the same file system name and comparing the file identifiers for the root directory.

**End of Programming Interface**

**End of Programming Interface**

## File Type and File Mode Values

**Programming Interface**

**Programming Interface**

A mode value is input to OS/390 UNIX System Services chmod, open, creat, mkdir, and umask, and output by OS/390 UNIX System Services stat and fstat. The mode value is defined as a mode_t data type and consists of a one-byte file type and three bytes for the file modes. The file mode specifies the permission bits and the S_ISUID, S_ISGID, and S_ISVTX bits for a file.

**End of Programming Interface**

**Programming Interface**

The OS/390 UNIX System Services macro BPXYMODE defines the mode_t values as:

**Programming Interface**

- Bits 0–7: file type, mapped by OS/390 UNIX System Services macro BPXYFTYP

**End of Programming Interface**

**Programming Interface**

- Bits 8–13: reserved

**End of Programming Interface**

**Programming Interface**

- Bits 14–31: available to the security product:

  **Programming Interface**

  – Bits 14–19: reserved

  **End of Programming Interface**

  **Programming Interface**

– Bit 20: S_ISUID (set user ID on execution)

**End of Programming Interface**

**Programming Interface**

– Bit 21: S_ISGID (set group name on execution)

**End of Programming Interface**

**Programming Interface**

– Bit 22: S_ISVTX (keep loaded executable in storage)

**End of Programming Interface**

**Programming Interface**

– Bits 23–25: S_IRWXU (owner class mask)

**Programming Interface**

- Bit 23: S_IRUSR (read permission)

**End of Programming Interface**

**Programming Interface**

- Bit 24: S_IWUSR (write permission)

**End of Programming Interface**

**Programming Interface**

- Bit 25: S_IXUSR (search (if directory) or execute (otherwise) permission)

**End of Programming Interface**

**End of Programming Interface**

**Programming Interface**

– Bits 26–28: S_IRWXG (group class mask)

**Programming Interface**

- Bit 23: S_IRGRP (read permission)

**End of Programming Interface**

**Programming Interface**

- Bit 24: S_IWGRP (write permission)

**End of Programming Interface**

**Programming Interface**

- Bit 25: S_IXGRP (search (if directory) or execute (otherwise) permission)

**End of Programming Interface**

**End of Programming Interface**

**Programming Interface**

– Bits 29–31: S_IRWXO (other class mask)

**Programming Interface**

- Bit 23: S_IROTH (read permission)

**End of Programming Interface**

**Programming Interface**

- Bit 24: S_IWOTH (write permission)

**End of Programming Interface**

**Programming Interface**

- Bit 25: S_IXOTH (search (if directory) or execute (otherwise) permission)

**End of Programming Interface**

**End of Programming Interface**

**End of Programming Interface**

**End of Programming Interface**

**Programming Interface**

The system call services pass the mode parameter from the caller of the system call to the RACF callable service or from the RACF callable service to the caller of the system call. The system call service can change the file type but does not change the file mode bits.

**End of Programming Interface**

**Programming Interface**

Some RACF callable services test the file type to determine if the file is a directory. The **makeFSP** service sets the file type to "directory" if the file is a directory and sets it to zero otherwise.

**End of Programming Interface**

## IPC Security Packet (IISP)

Interprocess communication (IPC) requires RACF to do authorization and permission checking. IPC facilities of the OS/390 UNIX system allow two or more distinct processes to communicate with each other. RACF protects this environment so that only those processes with the correct authority can communicate.

Interprocess communication consists of message queueing, semaphores, and shared memory segments used by application programs. Each function requires a security action by OS/390 UNIX, which RACF performs to allow a secure environment to exist.

The IPC security packet (IISP) contains data needed to make security decisions. It is built when a new ID for an IPC key is created and is saved in memory by the kernel. The IISP is used in place of a profile in the RACF database to contain information about the IPC key's owner and access rights.

The **makeISP** service initializes the IPC security packet (IISP) for a new IPC key with the creator's user and group identifiers (UID and GID), the owner's UID and GID, the mode bits, the IPC key, and the IPC ID.

The **ck_IPC_access** service determines whether the current process has the requested access to an IPC key. The IISP of the key is passed with this request. The **ck_IPC_access** service is called separately for each IPC key.

For the OS/390 UNIX IPC_SET command, the **R_IPC_ctl** service modifies the owner's UID, owner's GID, and mode bits in the IISP for the IPC key if the authority is correct. For the OS/390 UNIX IPC_RMID command, the **R_IPC_ctl** service checks the authority of the current process to determine whether the resource can be removed.

The IISP consists of two parts, the root and the extension. The root is mapped by macro IRRPIISP. The root contains a pointer to the extension, which is mapped by the OS/390 UNIX mapping macro BPXYIPCP. Other products can read the IISP for reporting purposes using the IRRPIISP and BPXYIPCP mapping macros.

The IISP root contains the following data:

- Control block ID
- Version number
- ALET of the IPCP
- Address of the IPCP (mapped by OS/390 UNIX System Services macro BPXYIPCP)
- IPC key
- IPC ID

For the mapping of the IPC security packet, see *OS/390 SecureWay Security Server RACF Data Areas*.

## Interprocess Communications Permission (BPXYIPCP)

```
         BPXYIPCP   ,
** BPXYIPCP: Interprocess Communications Permission
**  Used By: MCT, MGT, SCT, SGT, QCT, QGT
IPC_PERM             DSECT ,      Interprocess Communications
IPC_UID              DS    F      Owner's effective user ID
IPC_GID              DS    F      Owner's effective group name
IPC_CUID             DS    F      Creator's effective user ID
IPC_CGID             DS    F      Creator's effective group name
IPC_MODE             DS    XL4    Mode, mapped by BPXYMODE
IPC#LENGTH   EQU   *-IPC_PERM  Length of Interprocess Control block
* Key:
IPC_PRIVATE          EQU   0      Private key.
* Mode bits:                     Map over S_TYPE in BPXYMODE
IPC_CREAT            EQU   1      Create entry if key does not exist.
IPC_EXCL             EQU   2      Fail if key exists.
* Flag bits - semop, msgrcv, msgsnd:
IPC_NOWAIT           EQU   1      Error if request must wait.
* Control Command:
IPC_RMID             EQU   1      Remove identifier.
IPC_SET              EQU   2      Set options.
IPC_STAT             EQU   3      Access status.
* CONSTANTS WHICH MAP OVER BYTE S_TYPE, SEE BPXYMODE
** BPXYIPCP End
```

## IPC Security Credentials (CREI)

The IPC security credentials (CREI) structure is used in the OS/390 UNIX System Services IPC system to pass data from the kernel to RACF.

The CREI is built by the kernel, and is created for each system call entry to RACF.

The CREI contains:

- **User information:** a user type field that indicates whether the caller is a standard OS/390 UNIX System Services process known to RACF, or a system function that is not a process.

  Functions that accept a system caller process the request as if the caller is a superuser. If an audit record is written, the user OS/390 UNIX user identifier (UID) and OS/390 UNIX group identifier (GID) values in the record are set to -1.
- **Audit data:** data known by the kernel that needs to be passed through the IPC system to the RACF callable services for auditing. This data includes an **audit function code,** which identifies the system call being processed. The audit function codes are described in *OS/390 SecureWay Security Server RACF Data Areas*.
- **IPC key:** the key of the IPC service that is being checked.
- **IPC identifier:** the identifier of the IPC service that is being checked.

The CREI structure is mapped by the IRRPCREI mapping macro.

For the mapping of the CREI, see *OS/390 SecureWay Security Server RACF Data Areas*.

# Chapter 2. Callable Services Descriptions

This chapter describes the RACF callable services. The services appear in alphabetic order. Table 1 lists each callable service's intended users.

*Table 1. Intended Use of RACF Callable Services*

| Callable Service | For Use By |
|---|---|
| "ck_access (IRRSKA00): Check Access" on page 10 | OS/390 UNIX file system or OS/390 UNIX servers |
| "ck_file_owner (IRRSKF00): Check File Owner" on page 13 | OS/390 UNIX file system or OS/390 UNIX servers |
| "ck_IPC_access (IRRSKI00): Check IPC Access" on page 15 | MVS BCP or OS/390 UNIX task level processes |
| "ck_owner_two_files (IRRSC200): Check Owner of Two Files" on page 17 | OS/390 UNIX file system and OS/390 UNIX servers. |
| "ck_priv (IRRSKP00): Check Privilege" on page 19 | OS/390 UNIX file system, MVS BCP, or OS/390 UNIX servers |
| "ck_process_owner (IRRSKO00): Check Process Owner" on page 21 | MVS BCP or OS/390 UNIX task level processes |
| "clear_setid (IRRSCS00): Clear Set ID" on page 24 | OS/390 UNIX file system or OS/390 UNIX servers |
| "deleteUSP (IRRSDU00): Delete USP" on page 25 | MVS BCP or OS/390 UNIX servers |
| "getGMAP (IRRSGM00): Get GID-to-Group-Name Mapping" on page 27 | MVS BCP |
| "get_uid_gid_supgrps (IRRSGE00): Get UIDs, GIDs, and Supplemental Groups" on page 29 | OS/390 UNIX file system |
| "getUMAP (IRRSUM00): Get UID-to-User-ID Mapping" on page 32 | MVS BCP |
| "initUSP (IRRSIU00): Initialize USP" on page 48 | MVS BCP or OS/390 UNIX servers |
| "initACEE (IRRSIA00): Initialize ACEE" on page 34 | OS/390 kernel on behalf of servers that use pthread_security_np servers or __login, or MVS servers that do not use OS/390 UNIX services |
| "makeFSP (IRRSMF00): Make IFSP" on page 51 | OS/390 UNIX file system or OS/390 UNIX servers |
| "makeISP (IRRSMI00): Make IISP" on page 53 | MVS BCP or OS/390 UNIX task level processes |
| "make_root_FSP (IRRSMR00): Make Root IFSP" on page 55 | DFSMS/MVS or OS/390 UNIX servers |
| "query_file_security_options (IRRSQF00): Query File Security Options" on page 57 | OS/390 UNIX file system |
| "query_system_security_options (IRRSQS00): Query System Security Options" on page 59 | MVS BCP |
| "R_admin (IRRSEQ00): RACF Administration API" on page 61 | Tivoli |
| "R_audit (IRRSAU00): Provide an Audit Interface" on page 98 | OS/390 UNIX file system, MVS BCP, or OS/390 UNIX servers |
| "R_chaudit (IRRSCA00): Change Audit Options" on page 100 | OS/390 UNIX file system or OS/390 UNIX servers |
| "R_chmod (IRRSCF00): Change File Mode" on page 103 | OS/390 UNIX file system or OS/390 UNIX servers |

*Table 1. Intended Use of RACF Callable Services (continued)*

| Callable Service | For Use By |
|---|---|
| "R_chown (IRRSCO00): Change Owner and Group" on page 105 | OS/390 UNIX file system or OS/390 UNIX servers |
| "R_datalib (IRRSDL00): OCSF Data Library" on page 107 | MVS BCP or OS/390 UNIX servers |
| "R_dceauth (IRRSDA00): Check a User's Authority" on page 116 | MVS BCP |
| "R_dceinfo (IRRSDI00): Retrieve or Set User Fields" on page 119 | MVS BCP |
| "R_dcekey (IRRSDK00): Retrieve or Set a DCE Password" on page 123 | MVS BCP |
| "R_dceruid (IRRSUD00): Determine the ID of a Client" on page 126 | OS/390 UNIX servers or MVS BCP |
| "R_exec (IRRSEX00): Set Effective and Saved UIDs/GIDs" on page 130 | MVS BCP or OS/390 UNIX task level processes |
| "R_fork (IRRSFK00): Fork a Process" on page 132 | MVS BCP or OS/390 UNIX task level processes |
| "R_getgroups (IRRSGG00): Get/Set Supplemental Groups" on page 134 | MVS BCP or OS/390 UNIX servers |
| "R_getgroupsbyname (IRRSUG00): Get Groups by Name" on page 137 | MVS BCP |
| "R_IPC_ctl (IRRSCI00): Perform IPC Control" on page 139 | MVS BCP or OS/390 UNIX task level processes |
| "R_ptrace (IRRSPT00): Ptrace Authority Check" on page 155 | MVS BCP or OS/390 UNIX task level processes |
| "R_setegid (IRRSEG00): Set Effective GID, Set All GIDs" on page 157 | MVS BCP |
| "R_seteuid (IRRSEU00): Set Effective UID, Set All UIDs" on page 159 | MVS BCP |
| "R_setgid (IRRSSG00): Set Group Name" on page 160 | MVS BCP |
| "R_setuid (IRRSSU00): Set OS/390 UNIX user identifier (UID)" on page 162 | MVS BCP |
| "R_umask (IRRSMM00): Set File Mode Creation Mask" on page 169 | MVS BCP or OS/390 UNIX servers |
| "R_usermap (IRRSIM00): Map application user" on page 170 | OS/390 application servers |

**Note:** In a server environment, work can be processed for more than one user in an address space. Callable services marked for use by OS/390 UNIX servers provide task-level support for server applications. Callable services marked as having support for task level processes use task-level support when OS/390 UNIX has indicated in the task's ACEE that this is a task level process. All other callable services assume that there is only one user per address space and provide only address-space-level support.

# ck_access (IRRSKA00): Check Access

## Function

The **ck_access** service determines whether the current process has the requested access to the element (directory or file) of a pathname whose IFSP is passed. It is called separately for each element.

## Requirements

| | |
|---|---|
| **Authorization:** | Any PSW key in supervisor state |
| **Dispatchable unit mode:** | Task of OS/390 UNIX user or any task if system user type is specified |
| **Cross memory mode:** | PASN = HASN or PASN not = HASN |
| **AMODE:** | 31 |
| **RMODE:** | Any |
| **ASC mode:** | Primary or AR mode |
| **Recovery mode:** | SETFRR |
| **Serialization:** | Enabled for interrupts |
| **Locks:** | No locks held |
| **Control parameters:** | The parameter list and the work area must be in the primary address space. ALETs must be passed for all parameters except the work area. The words containing the ALETs must be in the primary address space. |

## RACF Authorization

1. If the audit function code in the CRED is access, the real OS/390 UNIX user identifier (UID) and OS/390 UNIX group identifier (GID) of the calling process are used on the authority checks. Otherwise, the effective UID and GID for the calling process are used.
2. If the calling user has auditor authority and the access requested is search, or the access requested is read for a directory, access is allowed. This lets an auditor set the auditor audit options on any file without requiring that the auditor be given search access rights to all directories.
3. If the CRED user type is system, IRRSKA00 allows any access except when the requested access is execute and no execute permission bits are set for the file. No UIDs are used in this case, because no process exists.
4. If the caller is not a superuser, the permission bits did not allow the requested access, *and* the audit function code is listed in Table 2, an authorization check is performed on the corresponding resource in the UNIXPRIV class. If the authorization check is successful, the caller is treated as a superuser.

*Table 2. UNIXPRIV class resource names used in ck_access*

| Audit function code | Resource name | Access required |
|---|---|---|
| OPEN (for read), OPENDIR, READLINK, STAT, REALPATH | SUPERUSER.FILESYS | READ |
| OPEN (for write) | SUPERUSER.FILESYS | UPDATE |
| LINK, MKDIR, RENAME, RMDIR, SYMLINK, UNLINK | SUPERUSER.FILESYS | CONTROL |

5. If the user being checked is a superuser, IRRSKA00 allows any access except when the requested access is execute and no execute permission bits are set for the file. The user is considered a superuser if the selected UID is 0 or if the ACEE indicates trusted or privileged authority.
6. If the user is not system and is not a superuser, the permission bits for the file are checked to see if the access requested is allowed. If the selected UID matches the owner UID of the file, the owner permission bits are checked. If the UIDs don't match, the owner GID of the file is checked against the selected

user GID and the user's supplemental group list GIDs. If any one matches, the group permission bits are checked. If the UIDs and GIDs don't match, the other permission bits are checked.

## Format

```
CALL IRRSKA00 (Work_area,
               ALET, SAF_return_code,
               ALET, RACF_return_code,
               ALET, RACF_reason_code,
               ALET, Requested_access_code,
               ALET, FSP,
               ALET, File_identifier,
               ALET, CRED,
               ALET, Name_flag
              )
```

## Parameters

**Work_area**
The name of a 1024-byte work area for SAF and RACF usage. The work area must be in the primary address space.

**ALET**
The name of a word containing the ALET for the following parameter. Each parameter must have an ALET specified. Each ALET can be different. The words containing the ALETs must be in the primary address space.

**SAF_return_code**
The name of a fullword in which the SAF router returns the SAF return code.

**RACF_return_code**
The name of a fullword in which the service routine stores the return code.

**RACF_reason_code**
The name of a fullword in which the service routine stores the reason code.

**Requested_access_code**
The name of a 1-byte field containing the requested access. The defined codes are:

**X'00'** no access

**X'01'** Execute access

**X'02'** Write access

**X'03'** Write and execute access

**X'04'** Read access

**X'05'** Read and execute access

**X'06'** Read and write access

**X'07'** Read, write, and execute access

**X'81'** Search access (against a directory)

**X'87'** Any access

**FSP**
The name of the IFSP for the file being accessed.

**File_Identifier**
The name of a 16-byte area containing a unique identifier of the file.

**CRED**
The name of the CRED structure for the current file system syscall. See the *OS/390 SecureWay Security Server RACF Data Areas* book.

**Name_flag**
The name of a byte indicating which pathname and file name is being checked. The byte contains one of these values:

**0** Use the CRED_name_flag to determine pathname being checked.

**1** The old (or only) name is being checked.

**2** The new name is being checked.

## Return and Reason Codes

IRRSKA00 returns the following values in the reason and return code parameters:

| SAF Return Code | RACF Return Code | RACF Reason Code | Explanation |
|---|---|---|---|
| 0 | 0 | 0 | The service was successful. |
| 4 | 0 | 0 | RACF is not installed. |
| 8 | 8 | 4 | The user is not authorized to access the file. |
| 8 | 8 | 32 | The CRED user type is not supported. |

## Usage Notes

1. This service is intended only for use by an OS/390 UNIX System Services file system and by OS/390 UNIX System Services servers. The service contains support for OS/390 UNIX System Services servers, but can not be directly invoked by an OS/390 UNIX System Services server.
2. The access checks performed are POSIX file permission checks defined in POSIX 1003.1.
3. If the audit function code in the CRED is access, no audit record is written. Access checking only tests whether a process would have access if it were running with its real UID. It does not give access to the file.
4. If the calling syscall is not access, an audit record is optionally written, depending on the audit options in effect for the system.

## Related Services

R_chmod, R_chown

# ck_file_owner (IRRSKF00): Check File Owner

## Function

The **ck_file_owner** service checks whether the calling process is a superuser or is the owner of the file represented by the input

## Requirements

| | |
|---|---|
| **Authorization:** | Any PSW key in supervisor state |
| **Dispatchable unit mode:** | Task of OS/390 UNIX user |
| **Cross memory mode:** | PASN = HASN or PASN not = HASN |
| **AMODE:** | 31 |
| **RMODE:** | Any |
| **ASC mode:** | Primary or AR mode |
| **Recovery mode:** | SETFRR |
| **Serialization:** | Enabled for interrupts |
| **Locks:** | No locks held |
| **Control parameters:** | The parameter list and the work area must be in the primary address space. ALETs must be passed for all parameters except the work area. The words containing the ALETs must be in the primary address space. |

## RACF Authorization

1. The ck_file_owner service checks whether the calling process is a superuser.
2. The ck_file_owner service checks whether the calling process is the owner of the file represented by the input IFSP.
3. A process is the owner of a file if the process's effective UID is equal to the file's owner UID.

## Format

```
CALL IRRSKF00 (Work_area,
               ALET, SAF_return_code,
               ALET, RACF_return_code,
               ALET, RACF_reason_code,
               ALET, FSP,
               ALET, File_identifier,
               ALET, CRED
              )
```

## Parameters

**Work_area**
The name of a 1024-byte work area for SAF and RACF usage. The work area must be in the primary address space.

**ALET**
The name of a word containing the ALET for the following parameter. Each parameter must have an ALET specified. Each ALET can be different. The words containing the ALETs must be in the primary address space. ALETs must be passed for all parameters except the work area. The words containing the ALETs must be in the primary address space.

**SAF_return_code**
The name of a full word in which the SAF router returns the SAF return code.

**RACF_return_code**
The name of a full word in which the service routine stores the return code.

**RACF_reason_code**
The name of a full word in which the service routine stores the reason code.

**FSP**
The name of the IFSP for the file to be checked.

**File_Identifier**
The name of a 16-byte area containing a unique identifier of the file.

**CRED**
The name of the CRED structure for the current file system syscall. See *OS/390 SecureWay Security Server RACF Data Areas*.

## Return and Reason Codes

IRRSKF00 returns the following values in the reason and return code parameters:

| SAF Return Code | RACF Return Code | RACF Reason Code | Explanation |
|---|---|---|---|
| 0 | 0 | 0 | The service was successful. |
| 4 | 0 | 0 | RACF is not installed. |
| 8 | 8 | 4 | The user is not a superuser or the file owner. |
| 8 | 8 | 12 | An internal error occurred during RACF processing. |
| 8 | 8 | 32 | The CRED user type is not supported. |

## Usage Notes

1. This service is intended only for use by an OS/390 UNIX System Services file system and by OS/390 UNIX System Services servers. The service contains support for OS/390 UNIX System Services servers, but can not be directly invoked by an OS/390 UNIX System Services server.
2. An audit record is optionally written, depending on the audit options in effect for the system.

## Related Services

None

---

# ck_IPC_access (IRRSKI00): Check IPC Access

## Function

The **ck_IPC_access** service determines whether the current process has the requested access to the interprocess communication (IPC) key or identifier whose IPC security packet (IISP) is passed.

## Requirements

**Authorization:** Any PSW key in supervisor state
**Dispatchable unit mode:** Task of OS/390 UNIX user/any task if system user type is specified
**Cross memory mode:** PASN = HASN or PASN not = HASN
**AMODE:** 31
**RMODE:** Any
**ASC mode:** Primary or AR mode
**Recovery mode:** None
**Serialization:** Enabled for interrupts

**Locks:** No locks held

**Control parameters:** The parameter list and the work area must be in the primary address space. ALETs must be passed for all parameters except the work area. The words containing the ALETs must be in the primary address space.

## RACF Authorization

1. The access checks performed are XPG4 IPC permission checks defined in XPG4 System Interfaces and Headers, as follows:
   - The effective OS/390 UNIX user identifier (UID) and OS/390 UNIX group identifier (GID) for the calling process is used for all access checks.
   - If the CREI user type is system, IRRSKI00 allows any access. No UIDs or GIDs are used in this case because no process exists.
   - If the user being checked is a superuser, IRRSKI00 allows any access. The user is considered a superuser if the selected UID is 0 or if the ACEE indicates trusted or privileged authority.
   - If the user is not system and is not a superuser, the permission bits for the IPC key are checked to see if the access requested is allowed. If the effective UID matches either the owner UID or creator's UID of the IPC key, the USER permission bits are checked. If the UIDs do not match, the owner GID and creator's GID of the IPC key are checked against the user's effective GID and the user's supplemental group list GIDs. If any one matches, the GROUP permission bits are checked. If the UIDs and GIDs don't match, the OTHER permission bits are checked.

## Format

```
CALL IRRSKI00 (Work_area,
               ALET, SAF_return_code,
               ALET, RACF_return_code,
               ALET, RACF_reason_code,
               ALET, Requested_access_code,
               ALET, ISP,
               ALET, CREDIPC
              )
```

## Parameters

**Work_area**
The name of a 1024-byte work area for SAF and RACF usage. The work area must be in the primary address space.

**ALET**
The name of a word containing the ALET for the following parameter. Each parameter must have an ALET specified. Each ALET can be different. The words containing the ALETs must be in the primary address space.

**SAF_return_code**
The name of a fullword in which the SAF router returns the SAF return code.

**RACF_return_code**
The name of a fullword in which the service routine stores the return code.

**RACF_reason_code**
The name of a fullword in which the service routine stores the reason code.

**Requested_access_code**
The name of a one-byte field containing the requested access. The defined codes are:

**X'00'** No access

**X'02'** Write access (or alter access)

**X'04'** Read access

**X'06'** Read and write access

**ISP**
The name of the IISP for the key being accessed.

**CREDIPC**
The name of the CREI structure for the current IPC system callable service. Use the CREI to determine the IPC identifier and IPC key being used. See *OS/390 SecureWay Security Server RACF Data Areas*.

## Return and Reason Codes

IRRSKI00 returns the following values in the reason and return code parameters:

| SAF Return Code | RACF Return Code | RACF Reason Code | Explanation |
|---|---|---|---|
| 0 | 0 | 0 | The service was successful. |
| 4 | 0 | 0 | RACF is not installed. |
| 8 | 8 | 4 | The user is not authorized to access the IPC mechanism. |
| 8 | 8 | 32 | CREI user type is not supported. |

## Usage Notes

1. This service is intended for use only by the MVS BCP.
2. An audit record is optionally written, depending on the audit options in effect for the system.
3. This service uses task level support when OS/390 UNIX has indicated in the task's ACEE that this is a task level process.

## Related Services

makeISP, R_IPC_ctl

---

# ck_owner_two_files (IRRSC200): Check Owner of Two Files

## Function

The **ck_owner_two_files** service checks whether the calling process is a superuser or is the owner of either of the file/directory, or directory/directory entry pair represented by input values FSP1 and FSP2.

## Requirements

**Authorization:** Any PSW key in supervisor state
**Dispatchable unit mode:** Task of OS/390 UNIX user
**Cross memory mode:** PASN = HASN or PASN not = HASN

| | |
|---|---|
| **AMODE:** | 31 |
| **RMODE:** | Any |
| **ASC mode:** | Primary or AR mode |
| **Recovery mode:** | SETFRR |
| **Serialization:** | Enabled for interrupts |
| **Locks:** | No locks held |
| **Control parameters:** | The parameter list and the work area must be in the primary address space. ALETs must be passed for all parameters except the work area. The words containing the ALETs must be in the primary address space. |

## RACF Authorization

1. A process is the owner of the file if the process's effective OS/390 UNIX user identifier (UID) is equal to the file's owner UID.
2. If the caller is not superuser nor the owner, and the audit function code is listed in Table 3, an authorization check is performed on the corresponding resource name in the UNIXPRIV class. If the authorization check is successful, the caller is treated as a superuser.

*Table 3. UNIXPRIV class resource names used in ck_owner_two_files*

| Audit function code | Resource name | Access required |
|---|---|---|
| RENAME, RMDIR, UNLINK | SUPERUSER.FILESYS | CONTROL |

## Format

```
CALL IRRSC200  (Work_area,
                ALET, SAF_return_code,
                ALET, RACF_return_code,
                ALET, RACF_reason_code,
                ALET, FSP1,
                ALET, FSP2,
                ALET, File_identifier_1,
                ALET, File_identifier_2,
                ALET, CRED
               )
```

## Parameters

**Work_area**
The name of a 1024-byte work area for SAF and RACF usage. The work area must be in the primary address space.

**ALET**
The name of a word containing the ALET for the following parameter. Each parameter must have an ALET specified. Each ALET can be different. The words containing the ALETs must be in the primary address space.

**SAF_return_code**
The name of a fullword in which the SAF router returns the SAF return code.

**RACF_return_code**
The name of a fullword in which the service routine stores the return code.

**RACF_reason_code**
The name of a fullword in which the service routine stores the reason code.

**FSP1**
The name of the IFSP for the first file, directory, or directory entry to be checked. If FSP1 is a file, FSP2 must be a directory. If FSP1 is a directory entry, FSP2 must be a directory.

**FSP2**
The name of the IFSP for the second file, directory, or directory to be checked. If FSP2 is a file, FSP1 must be a directory. If FSP2 is a directory entry, FSP1 must be a directory.

**File_identifier_1**
The name of a 16-byte area containing a unique identifier of the first file to be checked.

**File_identifier_2**
The name of a 16-byte area containing a unique identifier of the second file to be checked.

**CRED**
The name of the CRED structure for the current file system syscall. See *OS/390 SecureWay Security Server RACF Data Areas*.

## Return and Reason Codes

IRRSC200 returns the following values in the reason and return code parameters:

| SAF Return Code | RACF Return Code | RACF Reason Code | Explanation |
|---|---|---|---|
| 0 | 0 | 0 | The service was successful. |
| 4 | 0 | 0 | RACF is not installed. |
| 8 | 8 | 4 | The caller is not a superuser or the file owner. |
| 8 | 8 | 12 | An internal error occurred during RACF processing. |
| 8 | 8 | 32 | CRED user type is not supported. |

## Usage Notes

1. This service is intended only for use by an OS/390 UNIX System Services file system and by OS/390 UNIX System Services servers. The service contains support for OS/390 UNIX System Services servers, but cannot be directly invoked by an OS/390 UNIX System Services server.
2. An audit record is optionally written, depending on the audit options in effect for the system.

## Related Services

None

# ck_priv (IRRSKP00): Check Privilege

## Function

The **ck_priv** service checks whether the calling process is a superuser.

## Requirements

| | |
|---|---|
| **Authorization:** | Any PSW key in supervisor state |
| **Dispatchable unit mode:** | Task of OS/390 UNIX user |
| **Cross memory mode:** | PASN = HASN or PASN not = HASN |
| **AMODE:** | 31 |
| **RMODE:** | Any |
| **ASC mode:** | Primary or AR mode |
| **Recovery mode:** | SETFRR |
| **Serialization:** | Enabled for interrupts |
| **Locks:** | No locks held |
| **Control parameters:** | The parameter list and the work area must be in the primary address space. ALETs must be passed for all parameters except the work area. The words containing the ALETs must be in the primary address space. |

## RACF Authorization

1. A superuser is a user whose process has an effective UID of 0 or has RACF trusted or privileged authority.
2. If the caller is not superuser and the audit function code is listed in Table 4, an authorization check is performed on the corresponding resource name in the UNIXPRIV class. If the authorization check is successful, the caller is treated as a superuser.

*Table 4. UNIXPRIV class resource names used in ck_priv*

| Audit function code | Resource name | Access required |
|---|---|---|
| MOUNT(nosetuid), UNMOUNT(nosetuid), CHMOUNT(nosetuid), | SUPERUSER.FILESYS.MOUNT | READ |
| MOUNTSETUID, UNMOUNTSETU, CHMOUNT(setuid) | SUPERUSER.FILESYS.MOUNT | UPDATE |
| QUIESCE(nosetuid), UNQUIESCE(nosetuid) | SUPERUSER.FILESYS.QUIESCE | READ |
| QUIESCESETU, UNQUIESCESU | SUPERUSER.FILESYS.QUIESCE | UPDATE |
| PFSCTL | SUPERUSER.FILESYS.PFSCTL | READ |
| SETPRIORITY, NICE | SUPERUSER.SETPRIORITY | READ |
| VREGISTER | SUPERUSER.FILESYS.VREGISTER | READ |

## Format

```
CALL IRRSKP00 (Work_area,
               ALET, SAF_return_code,
               ALET, RACF_return_code,
               ALET, RACF_reason_code,
               ALET, Audit_function_code
              )
```

## Parameters

**Work_area**
The name of a 1024-byte work area for SAF and RACF usage. The work area must be in the primary address space.

**ALET**
The name of a word containing the ALET for the following parameter. Each parameter must have an ALET specified. Each ALET can be different. The words containing the ALETs must be in the primary address space.

**SAF_return_code**
The name of a fullword in which the SAF router returns the SAF return code.

**RACF_return_code**
The name of a fullword in which the service routine stores the return code.

**RACF_reason_code**
The name of a fullword in which the service routine stores the reason code.

**Audit_function_code**
The name of a fullword containing a function code identifying the system call function being processed. See *OS/390 SecureWay Security Server RACF Data Areas* for a list of the defined codes.

## Return and Reason Codes

IRRSKP00 returns the following values in the reason and return code parameters:

| SAF Return Code | RACF Return Code | RACF Reason Code | Explanation |
|---|---|---|---|
| 0 | 0 | 0 | The user is privileged. |
| 4 | 0 | 0 | RACF is not installed. |
| 8 | 8 | 4 | The user is not privileged. |
| 8 | 8 | 12 | An internal error occurred during RACF processing. |

## Usage Notes

1. This service is intended for use only by the MVS BCP, an OS/390 UNIX System Services file system, and by OS/390 UNIX System Services servers. The service contains support for OS/390 UNIX System Services servers, but can not be directly invoked by an OS/390 UNIX System Services server.
2. An audit record is written.

## Related Services

None

---

# ck_process_owner (IRRSKO00): Check Process Owner

## Function

The **ck_process_owner** service checks whether the calling process is the owner of the target process.

## Requirements

| | |
|---|---|
| **Authorization:** | Any PSW key in supervisor state |
| **Dispatchable unit mode:** | Task of OS/390 UNIX user |
| **Cross memory mode:** | PASN = HASN or PASN not = HASN |
| **AMODE:** | 31 |
| **RMODE:** | Any |
| **ASC mode:** | Primary or AR mode |
| **Recovery mode:** | SETFRR |
| **Serialization:** | Enabled for interrupts |
| **Locks:** | No locks held |
| **Control parameters:** | The parameter list and the work area must be in the primary address space. ALETs must be passed for all parameters except the work area. The words containing the ALETs must be in the primary address space. |

## RACF Authorization

1. For request types 2, 3, and 4, IRRSKO00 checks whether the caller has superuser authority or is the owner of the target process, and returns a return and reason code indicating the result.
2. The caller is an owner of a process if either the real or effective OS/390 UNIX user identifier (UID) of the calling process is equal to either the real or saved UID passed in the *Target_process_UIDs* parameter area.
3. If the caller is not superuser nor the process owner, and the request type is listed in Table 5, an authorization check is performed on the corresponding resource name in the UNIXPRIV class. If the authorization check is successful, the caller is treated as a superuser.

*Table 5. UNIXPRIV class resource names used in ck_process_owner*

| Request type | Resource name | Access required |
|---|---|---|
| 2 | SUPERUSER.PROCESS.KILL | READ |
| 3 | SUPERUSER.PROCESS.GETPSENT | READ |

## Format

```
CALL IRRSKO00 (Work_area,
               ALET, SAF_return_code,
               ALET, RACF_return_code,
               ALET, RACF_reason_code,
               ALET, Request_type,
               ALET, Target_process_UIDs,
               ALET, Target_PID,
               ALET, Signal_code
              )
```

## Parameters

**Work_area**
The name of a 1024-byte work area for SAF and RACF usage. The work area must be in the primary address space.

**ALET**
The name of a word containing the ALET for the following parameter. Each

parameter must have an ALET specified. Each ALET can be different. The words containing the ALETs must be in the primary address space.

**SAF_return_code**
The name of a fullword in which the SAF router returns the SAF return code.

**RACF_return_code**
The name of a fullword in which the service routine stores the return code.

**RACF_reason_code**
The name of a fullword in which the service routine stores the reason code.

**Request_type**
The address of a byte containing a request type. The defined types are:

- 1 - audit-only request from kill. It is used when a SIGCONT signal is being sent to a process in the same session as the signalling process.
- 2 - kill request
- 3 - getpsent request
- 4 - open_tty request

**Target_process_UIDs**
The address of a 3-word area containing the real, effective, and saved OS/390 UNIX user identifiers (UIDs) (in that order) for the target process.

**Target_PID**
The name of a fullword containing the PID of the target process.

**Signal_code**
The address of a word containing a code that identifies the type of signal being sent. This code is used only for auditing. The signal code values are defined in the OS/390 UNIX macro BPXYSIGH. This parameter is ignored for request type 3.

## Return and Reason Codes

IRRSKO00 returns the following values in the reason and return code parameters:

| SAF Return Code | RACF Return Code | RACF Reason Code | Explanation |
|---|---|---|---|
| 0 | 0 | 0 | The service was successful. |
| 4 | 0 | 0 | RACF is not installed. |
| 8 | 8 | 4 | The caller is not the owner of the target process. |
| 8 | 8 | 8 | The request type is not valid. |
| 8 | 8 | 12 | An internal error occurred during RACF processing. |

## Usage Notes

1. This service is intended only for use by the MVS BCP.
2. An audit record is optionally written, depending on the audit options in effect for the system.
3. This service uses task level support when OS/390 UNIX has indicated in the task's ACEE that this is a task level process.

### Related Services

None

---

## clear_setid (IRRSCS00): Clear Set ID

### Function

The **clear_setid** service clears the S_ISUID, S_ISGID, and S_ISVTX bits for the file passed as input.

### Requirements

| | |
|---|---|
| **Authorization:** | Any PSW key in supervisor state |
| **Dispatchable unit mode:** | Task of OS/390 UNIX user |
| **Cross memory mode:** | PASN = HASN or PASN not = HASN |
| **AMODE:** | 31 |
| **RMODE:** | Any |
| **ASC mode:** | Primary or AR mode |
| **Recovery mode:** | SETFRR |
| **Serialization:** | Enabled for interrupts |
| **Locks:** | No locks held |
| **Control parameters:** | The parameter list and the work area must be in the primary address space. ALETs must be passed for all parameters except the work area. The words containing the ALETs must be in the primary address space. |

### RACF Authorization

None

### Format

```
CALL IRRSCS00 (Work_area,
               ALET,SAF_return_code,
               ALET, RACF_return_code,
               ALET, RACF_reason_code,
               ALET, FSP,
               ALET, File_identifier,
               ALET, CRED
              )
```

### Parameters

**Work_area**
The name of a 1024-byte work area for SAF and RACF usage. The work area must be in the primary address space.

**ALET**
The name of a word containing the ALET for the following parameter. Each parameter must have an ALET specified. Each ALET can be different. The words containing the ALETs must be in the primary address space.

**SAF_return_code**
The name of a fullword in which the SAF router returns the SAF return code.

**RACF_return_code**
The name of a fullword in which the service routine stores the return code.

**RACF_reason_code**
The name of a fullword in which the service routine stores the reason code.

**FSP**
The name of the IFSP in which the S_ISUID, S_ISGID, and S_ISVTX bits are to be cleared.

**File_Identifier**
The name of a 16-byte area containing a unique identifier of the file.

**CRED**
The name of the CRED structure for the current file system syscall. See *OS/390 SecureWay Security Server RACF Data Areas*.

## Return and Reason Codes

IRRSCS00 returns the following values in the reason and return code parameters:

| SAF Return Code | RACF Return Code | RACF Reason Code | Explanation |
|---|---|---|---|
| 0 | 0 | 0 | The service was successful. |
| 4 | 0 | 0 | RACF is not installed. |
| 8 | 8 | 32 | The CRED user type is not supported. |

## Usage Notes

1. This service is intended only for use by an OS/390 UNIX System Services file system and by OS/390 UNIX System Services servers. The service contains support for OS/390 UNIX System Services servers, but can not be directly invoked by an OS/390 UNIX System Services server.
2. The caller is responsible for preserving the updated IFSP.
3. If either bit was on, an audit record is optionally written.

## Related services

R_chmod, R_exec

# deleteUSP (IRRSDU00): Delete USP

## Function

The **deleteUSP** service deletes the security environment for the calling process. The caller can continue as an MVS user, but is no longer an OS/390 UNIX process.

## Requirements

**Authorization:** Any PSW key in supervisor state
**Dispatchable unit mode:** Task of OS/390 UNIX user
**Cross memory mode:** PASN = HASN
**AMODE:** 31
**RMODE:** Any
**ASC mode:** Primary or AR mode
**Recovery mode:** SETFRR
**Serialization:** Enabled for interrupts

**Locks:** No locks held

**Control parameters:** The parameter list and the work area must be in the primary address space. ALETs must be passed for all parameters except the work area. The words containing the ALETs must be in the primary address space.

## RACF Authorization

None

## Format

```
CALL IRRSDU00 (Work_area,
               ALET, SAF_return_code,
               ALET, RACF_return_code,
               ALET, RACF_reason_code,
              )
```

## Parameters

**Work_area**
The name of a 1024-byte work area for SAF and RACF usage. The work area must be in the primary address space.

**ALET**
The name of a word containing the ALET for the following parameter. Each parameter must have an ALET specified. Each ALET can be different. The words containing the ALETs must be in the primary address space.

**SAF_return_code**
The name of a fullword in which the SAF router returns the SAF return code.

**RACF_return_code**
The name of a fullword in which the service routine stores the return code.

**RACF_reason_code**
The name of a fullword in which the service routine stores the reason code.

## Return and Reason Codes

IRRSDU00 returns the following values in the reason and return code parameters:

| SAF Return Code | RACF Return Code | RACF Reason Code | Explanation |
|---|---|---|---|
| 0 | 0 | 0 | The service was successful. |
| 4 | 0 | 0 | RACF is not installed. |

## Usage Notes

1. This service is intended only for use by the MVS BCP and by OS/390 UNIX System Services servers. This service can be directly invoked by an OS/390 UNIX System Services server.
2. An audit record is optionally written, depending on the audit options in effect for the system.

## Related Services

initUSP

---

# getGMAP (IRRSGM00): Get GID-to-Group-Name Mapping

## Function

The **getGMAP** service returns the OS/390 UNIX group identifier (GID) or group name corresponding to the input group name or GID, based on the setting of an input lookup flag.

## Requirements

| | |
|---|---|
| **Authorization:** | Any PSW key in supervisor state |
| **Dispatchable unit mode:** | Task of OS/390 UNIX user |
| **Cross memory mode:** | PASN = HASN |
| **AMODE:** | 31 |
| **RMODE:** | Any |
| **ASC mode:** | Primary or AR mode |
| **Recovery mode:** | ESTAE. Caller cannot have an FRR active. |
| **Serialization:** | Enabled for interrupts |
| **Locks:** | No locks held |
| **Control parameters:** | The parameter list and the work area must be in the primary address space. ALETs must be passed for all parameters except the work area. The words containing the ALETs must be in the primary address space. |

## RACF Authorization

None

## Format

```
CALL IRRSGM00 (Work_area,
               ALET, SAF_return_code,
               ALET, RACF_return_code,
               ALET, RACF_reason_code,
               ALET, Flag,
               ALET, GID,
               ALET, group_name
              )
```

## Parameters

**Work_area**
The name of a 1024-byte work area for SAF and RACF usage. The work area must be in the primary address space.

**ALET**
The name of a word containing the ALET for the following parameter. Each parameter must have an ALET specified. Each ALET can be different. The words containing the ALETs must be in the primary address space.

**SAF_return_code**
The name of a fullword in which the SAF router returns the SAF return code.

**RACF_return_code**
The name of a fullword in which the service routine stores the return code.

**RACF_reason_code**
The name of a fullword in which the service routine stores the reason code.

**Flag**
The name of a word containing the lookup option:

**X'00000000'**
search by OS/390 UNIX group identifier (GID), return group name

**X'00000001'**
search by group name, return GID

**GID**
The name of a fullword for a OS/390 UNIX group identifier (GID). The GID is either input or output in this word, depending on the flag parameter.

**Group_name**
The name of an 8-byte area for the group name. The group name is left-justified and padded with blanks and is either input or output in the area, depending on the flag parameter.

## Return and Reason Codes

IRRSGM00 returns the following values in the reason and return code parameters:

| SAF Return Code | RACF Return Code | RACF Reason Code | Explanation |
|---|---|---|---|
| 0 | 0 | 0 | The service was successful. |
| 4 | 0 | 0 | RACF is not installed. |
| 8 | 8 | 4 | If search by GID: GID is not defined. If search by group name: The current group's profile has no OMVS segment. |
| 8 | 8 | 8 | The group name is not defined. |
| 8 | 8 | 12 | An internal error occurred during RACF processing. |
| 8 | 8 | 16 | Recovery could not be established. |
| 8 | 8 | 20 | OMVS segment of the current group's profile has no GID. |
| 8 | 8 | 24 | The maximum number of file descriptors (OPEN MAX) are currently open in the calling process. **Note:** RACF does not issue this return code, but other security products may. |
| 8 | 8 | 28 | The maximum allowable number of files is currently open in the system. **Note:** RACF does not issue this return code, but other security products may. |

## Usage Notes

- This service is intended only for use by the MVS BCP.
- If getGMAP is given a group name as input and the corresponding GROUP profile has no OMVS segment, getGMAP checks the BPX.DEFAULT.USER profile in the FACILITY class. This profile may contain a group name in its application

data field that provides a default OMVS segment. If this default is found, its OS/390 UNIX group identifier (GID) is returned to the issuer of getGMAP.

- Check if any logrec entry has been created to ensure getGMAP service was being run successfully. Refer to *OS/390 SecureWay Security Server RACF Diagnosis Guide* for detailed logrec information.

## Related Services

None

---

# get_uid_gid_supgrps (IRRSGE00): Get UIDs, GIDs, and Supplemental Groups

## Function

The **get_uid_gid_supgrps** service gets the real, effective, and saved OS/390 UNIX user identifiers (UIDs) and OS/390 UNIX group identifiers (GIDs), and the supplemental groups from the USP.

Because the size of the supplemental group list varies, IRRSGE00 checks the input group count before putting supplemental GIDs in the grouplist area. See **Group_count** under "Parameters" on page 30 for more information.

The GIDs are not explicitly added to or deleted from the supplemental group list. A GID is in this list if the user was a member of the group when the user's ACEE was created through a RACROUTE REQUEST=VERIFY request and if the GID was assigned to the group before the **initUSP** service was performed for the process.

## Requirements

| | |
|---|---|
| **Authorization:** | Any PSW key in supervisor state |
| **Dispatchable unit mode:** | Task of OS/390 UNIX user |
| **Cross memory mode:** | PASN = HASN or PASN not = HASN |
| **AMODE:** | 31 |
| **RMODE:** | Any |
| **ASC mode:** | Primary or AR mode |
| **Recovery mode:** | SETFRR |
| **Serialization:** | Enabled for interrupts |
| **Locks:** | No locks held |
| **Control parameters:** | The parameter list and the work area must be in the primary address space. ALETs must be passed for all parameters except the work area. The words containing the ALETs must be in the primary address space. |

## RACF Authorization

None

## Format

```
CALL IRRSGE00 (Work_area,
               ALET, SAF_return_code,
               ALET, RACF_return_code,
               ALET, RACF_reason_code,
               ALET, RACF_work_area,
               ALET, User_key,
               ALET, Group_count,
               ALET, Grouplist,
               ALET, Number_of_GIDs,
               ALET, Output_UIDs,
               ALET, Output_GIDs
              )
```

## Parameters

**Work_area**
The name of a 1024-byte work area for SAF and RACF usage. The work area must be in the primary address space.

**ALET**
The name of a word containing the ALET for the following parameter. Each parameter must have an ALET specified. Each ALET can be different. The words containing the ALETs must be in the primary address space.

**SAF_return_code**
The name of a fullword in which the SAF router returns the SAF return code.

**RACF_return_code**
The name of a fullword in which the service routine stores the return code.

**RACF_reason_code**
The name of a fullword in which the service routine stores the reason code.

**RACF_work_area**
The name of a 1024-byte work area for RACF use.

**User_key**
The name of a byte containing the user's key. This key is used to store into the output grouplist area. The key is in the four high-order bits of the byte.

**Group_count**
The name of a word containing the number of OS/390 UNIX group identifier (GID) entries that can be stored in the *Grouplist* area. If *Group_count* is:

1. 0, the *Grouplist* area is not used. IRRSGE00 returns the total supplemental GID count of the current process in the *Number_of_GIDs* parameter.
2. Less than the total supplemental GID count:
   a. An error code is returned.
   b. The GIDs of the supplemental groups for the current process are put into the *Grouplist* area, which can only accommodate the number of GIDs specified in the *Group_count* parameter.
   c. The count of the supplemental GIDs actually placed in the *Grouplist* area is returned in the *Number_of_GIDs* parameter.
   d. The *Group_count* field is set to the total supplemental GID count of the current process.

      The supplemental groups in the *Grouplist* area are listed in the same order as the group connections shown in the output of the LISTUSER command.

3. Greater than or equal to the total supplemental OS/390 UNIX group identifier (GID) count:
   a. The GIDs of the supplemental groups for the current process are put into the *Grouplist* area.
   b. The supplemental GID count of the current process is put into the *Number_of_GIDs* parameter.

**Grouplist**
The name of an area in which the GIDs of the supplemental groups for a process are returned. The *Group_count* parameter indicates the number of entries this area can contain. The GIDs are returned as consecutive 4-byte entries.

**Number_of_GIDs**
The name of a word in which the number of GIDs put in the *Grouplist* area is returned.

**Output_UIDs**
The name of a 3-word area in which, respectively, the real, effective, and saved OS/390 UNIX user identifiers (UIDs) are returned.

**Output_GIDs**
The name of a 3-word area in which, respectively, the real, effective, and saved OS/390 UNIX group identifiers (GIDs) are returned.

## Return and Reason Codes

IRRSGE00 returns the following values in the reason and return code parameters:

| SAF Return Code | RACF Return Code | RACF Reason Code | Explanation |
|---|---|---|---|
| 0 | 0 | 0 | The service was successful. |
| 4 | 0 | 0 | RACF is not installed. |
| 8 | 8 | 4 | *Group_count* is less than the number of supplemental groups (see item 2 on page 30 under the **Group_count** parameter). |
| 8 | 8 | 8 | The grouplist address is not valid. |
| 8 | 8 | 12 | An internal error occurred during RACF processing. |

## Usage Notes

- This service is intended only for use by OS/390 UNIX System Services. The service which executes in the primary address space contains support that accesses the home address space task control block and address space control block for the requested data.
- In order to support multiple processes in one address space, this function needs to return the requested data from either the task control area or the address space control area. The task control area is accessed before the address space control area.

## Related Services

None.

# getUMAP (IRRSUM00): Get UID-to-User-ID Mapping

## Function

The **getUMAP** service returns the OS/390 UNIX user identifier (UID) or user ID corresponding to the input user ID or UID, based on the setting of an input lookup flag.

## Requirements

| | |
|---|---|
| **Authorization:** | Any PSW key in supervisor state |
| **Dispatchable unit mode:** | Task of OS/390 UNIX user |
| **Cross memory mode:** | PASN = HASN |
| **AMODE:** | 31 |
| **RMODE:** | Any |
| **ASC mode:** | Primary or AR mode |
| **Recovery mode:** | ESTAE. Caller cannot have an FRR active. |
| **Serialization:** | Enabled for interrupts |
| **Locks:** | No locks held |
| **Control parameters:** | The parameter list and the work area must be in the primary address space. ALETs must be passed for all parameters except the work area. The words containing the ALETs must be in the primary address space. |

## RACF Authorization

None

## Format

```
CALL IRRSUM00 (Work_area,
               ALET, SAF_return_code,
               ALET, RACF_return_code,
               ALET, RACF_reason_code,
               ALET, Flag,
               ALET, UID,
               ALET, Userid
              )
```

## Parameters

**Work_area**
The name of a 1024-byte work area for SAF and RACF usage. The work area must be in the primary address space.

**ALET**
The name of a word containing the ALET for the following parameter. Each parameter must have an ALET specified. Each ALET can be different. The words containing the ALETs must be in the primary address space.

**SAF_return_code**
The name of a fullword in which the SAF router returns the SAF return code.

**RACF_return_code**
The name of a fullword in which the service routine stores the return code.

**RACF_reason_code**
The name of a fullword in which the service routine stores the reason code.

**Flag**
The name of a word containing the lookup option:

**X'00000000'**
search by OS/390 UNIX user identifier (UID), return user ID

**X'00000001'**
search by user ID, return OS/390 UNIX user identifier (UID)

**UID**
The name of a fullword for a OS/390 UNIX user identifier (UID). The UID is either input or output in this word, depending on the flag parameter.

**Userid**
The name of an 8-byte area for the user ID. The user ID is left-justified and padded with blanks, and is either input or output in the area depending on the flag parameter.

## Return and Reason Codes

IRRSUM00 returns the following values in the reason and return code parameters:

| SAF Return Code | RACF Return Code | RACF Reason Code | Explanation |
|---|---|---|---|
| 0 | 0 | 0 | The service was successful. |
| 4 | 0 | 0 | RACF is not installed. |
| 8 | 8 | 4 | If search by UID: UID is not defined. If search by user ID: The user's profile has no OMVS segment. |
| 8 | 8 | 8 | User ID is not defined. |
| 8 | 8 | 12 | An internal error occurred during RACF processing |
| 8 | 8 | 16 | Recovery could not be established. |
| 8 | 8 | 20 | The OMVS segment of the user's profile has no UID. |
| 8 | 8 | 24 | The maximum number of file descriptors (OPEN MAX) are currently open in the calling process. **Note:** RACF does not issue this return code, but other security products may. |
| 8 | 8 | 28 | The maximum allowable number of files is currently open in the system. **Note:** RACF does not issue this return code, but other security products may. |

## Usage Notes

- This service is intended only for use by the MVS BCP.
- If getUMAP is given a user ID as input, and the corresponding USER profile has no OMVS segment, getUMAP checks the BPX.DEFAULT.USER profile in the BPX.DEFAULT.USER profile in the FACILITY class. This profile may contain a user ID in its application data field that provides a default OMVS segment. If this default is found, its UID is returned to the issuer of getUMAP.

- Check if any logrec entry has been created to ensure getUMAP service was being run successfully. Refer to *OS/390 SecureWay Security Server RACF Diagnosis Guide* for detailed logrec information.

## Related Services

None.

# initACEE (IRRSIA00): Initialize ACEE

## Function

The **initACEE** service provides an interface for creating and managing RACF security contexts through the OS/390 UNIX System Services pthread_security_np service, __login service, or by other MVS server address spaces that do not use OS/390 UNIX services. This service also provides an interface for registering and deregistering certificates through the OS/390 UNIX System Services __security service. It also provides an interface for querying a certificate to determine if it is associated with a user ID.

## Requirements

| | |
|---|---|
| **Authorization:** | Any PSW key in supervisor state |
| **Dispatchable unit mode:** | Task of user |
| **Cross memory mode:** | PASN = HASN |
| **AMODE:** | 31 |
| **RMODE:** | Any |
| **ASC mode:** | Primary or AR mode |
| **Recovery mode:** | ESTAE. Caller cannot have an FRR active. |
| **Serialization:** | Enabled for interrupts |
| **Locks:** | No locks held |
| **Control parameters:** | The parameter list and the work area must be in the primary address space. The words containing the ALETs must be in the primary address space. |

## Linkage Conventions

The parameter list for this callable service is intended to be variable length to allow for future expansion. To allow for this, the last word in the parameter list must have a 1 in the high-order (sign) bit.

## RACF Authorization

1. If the function_code indicates that a certificate is to be registered or deregistered, initACEE will perform the following authority checks:
   - To register a certificate with the current user ID, the caller must be RACF SPECIAL or have at least READ authority to the IRR.DIGTCERT.ADD resource in the FACILITY class.
   - To deregister a certificate with the current user ID, the caller must be RACF SPECIAL or have at least READ authority to the IRR.DIGTCERT.DELETE resource in the FACILITY class.
   - To register a certificate as a CERTAUTH certificate, the caller must be RACF SPECIAL or have at least CONTROL authority to the IRR.DIGTCERT.ADD resource in the FACILITY class.

2. If the function_code indicates that an ACEE is to be created or a certificate is to be queried and the service determines that the user ID to use is specified in the hostIdMappings extension of the input certificate, the caller's authority to the IRR.HOST.*(host-name)* resource in the SERVAUTH class is checked. (The value for *host-name* is specified in the hostIdMappings extention.) The resource must exist and the caller must have READ authority to it, otherwise the extension is ignored.

**Note:** To determine the caller, the current TCB is checked for an ACEE. If one is found, the authority of that user is checked. If there is no ACEE associated witht the current TCB, the ACEE associated with the address space is used to locate the user ID.

## Format

```
CALL IRRSIA00 (Work_area,
               ALET, SAF_return_code,
               ALET, RACF_return_code,
               ALET, RACF_reason_code,
                     Function_code,
                     Attributes,
                     RACF_userid,
                     ACEE_ptr,
                     APPL_id,
                     Password,
                     Logstring,
                     Certificate,
                     ENVR_in,
                     ENVR_out,
                     Output_area
                     X500name,
                     Variable_list
                 )
```

## Parameters

**Work_area**
The name of a 1024-byte work area for SAF and RACF usage. The work area must be in the primary address space.

**ALET**
The name of a word containing the ALET for the following parameter. Each ALET can be different. The words containing the ALETs must be in the primary address space.

**SAF_return_code**
The name of a fullword in which the SAF router returns the SAF return code.

**RACF_return_code**
The name of a fullword in which the service routine stores the return code.

**RACF_reason_code**
The name of a fullword in which the service routine stores the reason code.

**Function_code**
The name of a 1 byte area containing the function code

**X'01'** Create an ACEE.

**X'02'** Delete an ACEE.

**X'03'** Purge all managed ACEEs.

**X'04'** Register a certificate

**X'05'** Deregister a certificate

**X'06'** Query a certificate

**Attributes**
The name of a 4 byte area containing information about the function to be performed. Zero or more attributes can be set. (See Table 6 on page 39 for the values allowed for the Attributes parameter.)

**RACF_userid**
The name of a 9 byte area that consists of a 1 byte length field followed by up to 8 characters. It must be specified in upper case. If not specified, the length must equal 0.

**ACEE_ptr**
The name of a 4 byte area that contains the ACEE address.

**APPL_id**
The name of a 9 byte area that consists of a 1 byte length field followed by the name of the application to be used if verifying the user's authority to access the application. This saves the application from having to do a separate authorization check. When using certificate mapping profiles, the application name is also used as part of the additional criteria in determining a user ID when a certificate is passed to initACEE. It must be specified in upper case. If not specified, the length must equal zero.

**Password**
The name of a 9 byte area that consists of a 1 byte length field followed by the password or PassTicket provided by the user. It must be specified in upper case. If not specified, the length must equal zero.

**Logstring**
The name of an area that consists of a 1 byte length field followed by character data to be written to the system-management-facilities (SMF) data set, together with any RACF audit information, if logged. If not specified, the length must equal zero.

**Certificate**
The name of an area that consists of a 4 byte length field followed by a digital certificate. If not specified, the length must equal 0; or the end of the parameter list must be indicated by the setting of the high order bit in the address of the previous parameter. The certificate must be a single DER encoded X.509 certificate. For the registration and deregistration functions, PKCS #7, PEM, or Base64 encoded certificates are also allowed.

**ENVR_in**
The name of the data structure that contains the information necessary to re-create a security environment. The data structure must have the format shown in Table 7 on page 39. See the ENVR_out parameter for additional information on this data structure and the ENVR object to which it points. The structure must reside on a doubleword boundary.

While the format of the data structure pointed to by ENVR_in is known to the initACEE invokers, the content of the object itself is determined by the external security product.

The input for this parameter can be the output from a previous initACEE with the ENVR_out parameter specified, or from RACROUTE REQUEST=VERIFY or REQUEST=EXTRACT, with the ENVROUT parameter specified.

If ENVR_in is not specified, the ENVR object length must equal 0, or the end of the parameter list must be indicated by the setting of the high order bit in the address of a previous parameter. ENVR_in should not be specified when requesting that an ENVR object be returned (INTA_ENVR_RET).

For more information about the ENVR data structure, see *OS/390 SecureWay Security Server External Security Interface (RACROUTE) Macro Reference*.

**ENVR_out**
The name of the data structure to contain the security environment that was just created. The data structure must have the format shown in Table 7 on page 39. This data structure describes the storage location for the ENVR object that is created as part of this initACEE create request.

While the format of the data structure pointed to by ENVR_out itself is known to the initACEE invokers, the content of the object itself is determined by the external security product.

The ENVR object storage area can be supplied by the caller or obtained by RACF. If supplied by the caller, it must be on a doubleword boundary and be associated with the job step task. If RACF obtains the storage area, it is on a doubleword boundary and is associated with the job step task. The storage is allocated based on the mode of the caller (LOC=ANY for 31-bit callers and LOC=24 for 24-bit callers).

Storage for the ENVR object is obtained and freed in the subpool and key specified by the caller in the ENVR_out data structure. For additional details on specifying the ENVR object storage area length and address, see Table 8 on page 39.

Since the ENVR object length is returned to the caller, the ENVR object can be moved from one storage area to another. It is intended for use on subsequent initACEEs with the ENVR_in parameter, or on RACROUTE REQUEST=VERIFY with the ENVRIN parameter, as input when rebuilding a user's security environment. It should not be saved for a long period or passed to another system that does not share the same RACF database.

If the Attributes parameter indicates that an ENVR object should be returned (INTA_ENVR_RET), then this parameter must be specified with at a minimum the values for the subpool and key fields.

For more information about the ENVR data structure, see *OS/390 SecureWay Security Server External Security Interface (RACROUTE) Macro Reference*.

**Output_area**
The name of a fullword in which the service routine stores the address of an area containing data about the user. The output area is obtained in the primary address space, in subpool 229, and must be freed by the caller of initACEE. The following data is returned; the area returned is mapped my macro IRRPOUSP (OUSP):

- TSO user ID
- OS/390 UNIX user identifier (UID) of user
- OS/390 UNIX group identifier (GID) of current group
- Home directory path name
- Initial program path name

- User limits (when OUSP version is greater than 0)

If the Attributes parameter indicates that an OUSP should be returned (INTA_USP and INTA_OUSP_RET), then this parameter must be specified. If the Attributes do not indicate that an OUSP should be returned, then the fullword must equal 0, or the end of the parameter list must be indicated by the setting of the high order bit in the address of a previous parameter.

**X500name**
The name of a fullword in which the service routine stores the address of the X500 name pair data structure if the function code indicates a certificate is being queried, and the attributes indicate that an X500 name pair should be returned. The X500 name pair data structure is obtained in the primary address space, in subpool 229, and must be freed by the caller of initACEE.

If the function code indicates that an ACEE is to be created, and the RACF_userid parameter is specified, X500name can supply the name of a fullword containing the address of the X500 name pair to be associated with the ACEE. The X500 name pair should previously have been obtained, along with the RACF user ID, by querying a certificate using initACEE. Both the issuer's name and subject's name must be supplied, and the length of each must be in the range 1 to 255 to prevent a parameter list error. If a valid X500 name pair is supplied, the ACEE created will point to a copy of the name pair, and it will subsequently be used in auditing.

**Variable_list**
The name of the data structure that contains the additional criteria to be used to determine the user ID associated with the certificate supplied to initACEE. The criteria value data structure is a 4-byte number of value entries, followed by that number of entries. Each value entry consists of an 8-byte value name, a 4-byte value length, and the value. The value name must be padded on the right with blanks if it is less than 8-bytes. The value length must be in the range of 1 to 255. If it is outside of this range, a parameter list error will result. A maximum of 10 values may be specified. If the number of values is greater than 10, a parameter list error will result.

Variable names should be meaningful to the caller of initACEE. Making the 3 character prefix associated with the product calling initACEE part of the variable name will ensure that it is unique. For example, assume RACF implemented a server that calls initACEE for its clients. It will pass a variable, IRRSLVL, which has 2 values. The values are LOW and HIGH. LOW is if the user is accessing the server from the internet and HIGH is if the user is accessing the server from the intranet. The variable_list containing the variable name and its value, LOW or HIGH, is passed to initACEE, along with the certificate supplied by the user. The value of the variable will be used as additional criteria in selecting which user ID the certificate maps to. All callers of initACEE should document their variable names, and the values they pass for each name, in their product documentation.

All value names and values should be upper case. Do not specify the APPLID or SYSID criteria values in the variable_list. These are determined from the APPL_id parameter and MVS control; blocks, respectively. If they are specified in the variable_list, that specification will be ignored.

This parameter is ignored unless the certificate parameter is specified, and the function code indicates that an ACEE is to be created, or that the certificate is to be queried to find a user ID. If the certificate is defined to RACF in the

DIGTCERT class, additional criteria will not be used, and the variable_list values will be ignored. If the certificate is not defined in the DIGTCERT class, the values in the variable list will be used along with APPL_id and SYSID to look for an associated user ID using the DIGTNMAP and DIGTCRIT classes if these classes are active and have been RACLISTed with SETROPTS.

*Table 6. Values Allowed for Attributes Parameter*

| | | |
|---|---|---|
| INTA_MANAGED | X'80000000' | Create an ACEE for the user ID that is cached by RACF. |
| INTA_USP | X'40000000' | Create a USP for the user ID |
| INTA_TASK_LVL | X'20000000' | For create function code, create an ACEE and attach to the current TCB. For delete function code, delete the ACEE attached to the current TCB. |
| INTA_UNAUTH_CLNT | X'10000000' | Create an ACEE for an unauthenticated client. |
| INTA_AUTH_CLNT | X'08000000' | Create an ACEE for an authenticated client. |
| INTA_MGS_SUPP | X'04000000' | Suppress RACF messages produced as a result of creating a user's security context. |
| INTA_ENVR_RET | X'02000000' | Return an ENVR object for the ACEE created by this request. |
| INTA_NO_TIMEOUT | X'01000000' | Create a managed ACEE that does not time out. When this bit and INTA_MANAGED are set for the creation of a new managed ACEE, the ACEE is cached and does not expire after 5 minutes. |
| INTA_OUSP_RET | X'00800000' | Return an OUSP in the output area |
| INTA_X500_RET | X'00400000' | Return an X500 name pair |

*Table 7. ENVR Data Structure*

| Description | Length (Bytes) | ENVR_out Usage | ENVR_in Usage |
|---|---|---|---|
| ENVR object length | 4 | Output | Input |
| ENVR object storage area length | 4 | Input/Output (see Table 8) | Input |
| ENVR object storage area address | 4 | Input/Output (see Table 8) | Input |
| ENVR object storage area subpool | 1 | Input | n/a |
| ENVR object storage area key | 1 | Input | n/a |

*Table 8. ENVR_out Storage Area Processing*

| ENVR Object Storage Area Length | ENVR Object Storage Area Address | Result |
|---|---|---|

*Table 8. ENVR_out Storage Area Processing (continued)*

| | | |
|---|---|---|
| Zero | Any value | RACF obtains storage size needed to contain ENVR object and sets ENVR object storage area length and address fields. |
| Nonzero | Zero | RACF obtains storage size specified or minimum needed to contain ENVR object and sets ENVR object storage area length and address fields. |
| Nonzero | Nonzero | RACF uses the area provided if large enough to contain ENVR object. If too small, RACF freemains the area, obtains a larger area, and sets ENVR object storage area length and address fields. |

*Table 9. X500 Name Pair Data Structure*

| Offset | Length(Bytes) | Description |
|---|---|---|
| 0 | 4 | Length of name pair data structure |
| 4 | 2 | Length of issuer's name (1 to 255) |
| 6 | 2 | Length of subject's name (1 to 255) |
| 8 | 1 to 255 | Issuer's distinguished name |
| * | 1 to 255 | Subject's distinguished name |

*Table 10. Criteria Value Data Structure*

| Offset | Length(Bytes) | Description |
|---|---|---|
| 0 | 4 | Number of value entries |
| 4 | 8 | Value name |
| 12(C) | 4 | Value length (1 to 255) |
| 16(10) | 1 to 255 | Value |

## Return and Reason Codes

IRRSIA00 returns the following values in the reason and return code parameters:

*Table 11. initACEE Create Return Codes*

| SAF Return Code | RACF Return Code | RACF Reason Code | Explanation |
|---|---|---|---|
| 0 | 0 | 0 | The service was successful. |
| 4 | 0 | 0 | RACF is not installed. |
| 8 | 8 | 4 | Parameter list error occurred. |

*Table 11. initACEE Create Return Codes (continued)*

| SAF Return Code | RACF Return Code | RACF Reason Code | Explanation |
|---|---|---|---|
| 8 | 8 | 8 | An internal error occurred during RACF processing. |
| 8 | 8 | 12 | Recovery environment could not be established. |
| 8 | 8 | 16 | User ID is not defined to RACF. |
| 8 | 8 | 20 | Password or Pass Ticket is not valid. |
| 8 | 8 | 24 | Password is expired. |
| 8 | 8 | 28 | User ID is revoked. |
| 8 | 8 | 32 | User is not authorized. |
| 8 | 8 | 36 | Certificate is not valid. |
| 8 | 8 | 40 | Either no user ID or userid mapping is defined for this certificate or the status of the certificate or mapping is NOTRUST, or there are no mapping profiles associated with the certificate. Mapping profiles can either be defined as a Certificate Name Filtering profile or as SERVAUTH profiles that are used for HostID Mappings. See Usage Note number 37. |
| 8 | 12 | InitUSP reason code | initUSP failed. See initUSP reason codes in "Return and Reason Codes" on page 49. |

*Table 12. initACEE Delete Return Codes*

| SAF Return Code | RACF Return Code | RACF Reason Code | Explanation |
|---|---|---|---|
| 0 | 0 | 0 | The service was successful. |
| 4 | 0 | 0 | RACF is not installed. |
| 8 | 8 | 4 | Parameter list error occurred. |
| 8 | 8 | 8 | An internal error occurred during RACF processing. |
| 8 | 8 | 12 | Recovery environment could not be established. |
| 8 | 8 | 16 | An attempt was made to delete the server address space ACEE before invoking initACEE to purge all managed ACEEs. |

*Table 13. initACEE Purge Return Codes*

| SAF Return Code | RACF Return Code | RACF Reason Code | Explanation |
|---|---|---|---|
| 0 | 0 | 0 | The service was successful. |
| 4 | 0 | 0 | RACF is not installed. |
| 8 | 8 | 4 | Parameter list error occurred. |
| 8 | 8 | 8 | An internal error occurred during RACF processing. |

*Table 13. initACEE Purge Return Codes (continued)*

| SAF Return Code | RACF Return Code | RACF Reason Code | Explanation |
|---|---|---|---|
| 8 | 8 | 12 | Recovery environment could not be established. |
| 8 | 8 | 16 | There are managed ACEEs that are still in use. |

*Table 14. initACEE Register and Deregister Return Codes*

| SAF Return Code | RACF Return Code | RACF Reason Code | Explanation |
|---|---|---|---|
| 0 | 0 | 0 | The service was successful. |
| 4 | 0 | 0 | RACF is not installed. |
| 8 | 8 | 4 | Parameter list error occurred. |
| 8 | 8 | 8 | An internal error occurred during RACF processing |
| 8 | 8 | 12 | Recovery environment could not be established. |
| 8 | 8 | 16 | The user is not authorized. |
| 8 | 8 | 20 | The certificate does not meet RACF requirements. |
| 8 | 8 | 24 | The certificate is defined for another user. |
| 8 | 8 | 28 | RESERVED |
| 8 | 8 | 32 | RESERVED |
| 8 | 8 | 36 | The certificate is not valid. |

*Table 15. initACEE Query Return Codes*

| SAF Return Code | RACF Return Code | RACF Reason Code | Explanation |
|---|---|---|---|
| 0 | 0 | 0 | The service was successful. |
| 4 | 0 | 0 | RACF is not installed. |
| 8 | 8 | 4 | Parameter list error occurred. |
| 8 | 8 | 8 | An internal error occurred during RACF processing. |
| 8 | 8 | 12 | Recovery environment could not be established. |
| 8 | 8 | 16 | RESERVED |
| 8 | 8 | 20 | RESERVED |
| 8 | 8 | 24 | RESERVED |
| 8 | 8 | 28 | RESERVED |
| 8 | 8 | 32 | RESERVED |

*Table 15. initACEE Query Return Codes (continued)*

| SAF Return Code | RACF Return Code | RACF Reason Code | Explanation |
|---|---|---|---|
| 8 | 8 | 36 | This certificate is not valid. |
| 8 | 8 | 40 | No user ID is defined for this certificate. |

*Table 16. Parameter Usage*

| Parameter | Create ACEE | Delete ACEE | Purge ACEE | Reg/Dereg Certificate | Query Certificate |
|---|---|---|---|---|---|
| SAF_return_code | Output | Output | Output | Output | Output |
| RACF_return_code | Output | Output | Output | Output | Output |
| RACF_reason_code | Output | Output | Output | Output | Output |
| Function_code | Input | Input | Input | Input | Input |
| Attributes | Input | Input | n/a | n/a | Input |
| RACF_userid | Input | n/a | n/a | n/a | Output |
| ACEE_ptr | Output | Input | n/a | n/a | n/a |
| APPL_id | Input | n/a | n/a | n/a | n/a |
| Password | Input | n/a | n/a | n/a | n/a |
| Logstring | Input | n/a | n/a | n/a | n/a |
| Certificate | Input | n/a | n/a | Input | Input |
| ENVR_in | Input | n/a | n/a | n/a | n/a |
| ENVR_out | Input/ Output See Table 7 on page 39 | n/a | n/a | n/a | n/a |
| Output_area | Output | n/a | n/a | n/a | n/a |
| X500name | Input | n/a | n/a | n/a | Output |
| Variable_list | Input | n/a | n/a | n/a | Input |

## Usage Notes

1. This service is only intended for use by the OS/390 UNIX kernel or by other MVS servers that do not use OS/390 UNIX System Services.
2. This service can only be used by supervisor state callers.
3. An ALET must be specified for the SAF_return_code, RACF_return_code, and RACF_reason_code parameters.
4. When ACEEs are created by initACEE, the following information is used on the RACROUTE REQUEST=VERIFY:
   - Password, if verifying a password
   - Appl_id, if verifying authority to an application
   - Logstring, if any audit records are created as a result of authenticating the user ID
   - LOC=ANY. If the caller is running in 31-bit address mode, the ACEE may be allocated above the 16MB line.
   - Subpool of the address space ACEE is used for the SUBPOOL keyword

- ENVROUT, if an ENVR object data structure address was supplied by the ENVR_out parameter.
- X500name, if the RACF_userid and X500_name parameters were specified, or if a certificate was provided as input and an associated user ID was found using the DIGTNMAP class profiles.

5. When creating an ACEE, statistics are updated on the first request per day for each user ID.
6. Audit records are written only in the following situations:
   a. An ACEE is to be created and a password has been specified that is not the user's current password.
   b. An ACEE is to be created and a PassTicket has been specified that does not evaluate.
   c. An ACEE is to be created and the user ID has been revoked.
   d. A certificate is to be registered, and the user is not authorized to the FACILITY class resource IRR.DIGTCERT.ADD.
   e. A certificate is to be deregistered and the user is not authorized to the FACILITY class resource IRR.DIGTCERT.DELETE.
   f. A certificate is successfully registered or deregistered, and SETROPTS AUDIT(USER) is in effect, or UAUDIT is in effect for the user, or the user has SPECIAL authority and SETROPTS SAUDIT is in effect.
   g. An ACEE is to be created and a certificate has been specified that does not correspond to a RACF user ID.
   h. An ACEE is to be created and a certificate has been specified that is not trusted.
7. If an ACEE is to be anchored off the current TCB, then the INTA_TASK_LVL attribute must be set. Any value passed in ACEE_ptr is ignored, and the ACEE address is not returned. If an ACEE address is to be returned, the INTA_TASK_LVL attribute must be off. This results in the ACEE address being returned in the ACEE_ptr parameter area.
8. If an ACEE is to be deleted from the current TCB, then the INTA_TASK_LVL attribute must be set. If this is not done, the ACEE_ptr parameter must point to the address of the ACEE to be deleted.
9. If the function_code and attributes indicate that an ACEE is to be created and anchored off the TCB and there is an ACEE already anchored off the TCB, the caller receives a parameter list error.
10. If the function_code and attributes indicate that an ACEE is to be deleted from the TCB and there is no ACEE anchored to the TCB, the caller receives a parameter list error.
11. If the last word in the parameter list does not have a 1 in the high-order (sign) bit, the caller receives a parameter list error. The first parameter that can have the high-order bit on, ending the parameter list, is the logstring parameter
12. When the application is terminating and there are no tasks outstanding, initACEE should be called to purge all the managed ACEEs. Then the ACEE for the application server address space can be deleted.
13. The RACROUTE service should not be used to delete the managed ACEEs.
14. You can find parameter usages in Table 16 on page 43.
15. The service serializes resources at the address space level with a STEP ENQ on QNAME "SYSZRACF".
16. If the function_code indicates that an ACEE is to be created and the length of the certificate parameter is not zero, then the length of the RACF_user ID and

password must both be 0. If a RACF_user ID or password is supplied with the certificate, the caller receives a parameter list error.

17. If the function_code indicates that an ACEE is to be deleted or that managed ACEEs should be purged, and the length of the certificate parameter is not zero, then the caller receives a parameter list error.
18. If the function_code indicates that a certificate is to be registered, deregistered, or queried, and the length of the certificate parameter is zero, then the caller receives a parameter list error.
19. The certificate supplied by the certificate parameter is used only to identify a RACF user ID. It is expected that the certificate was previously verified. Note the following additional details regarding initACEE's certificate processing:
    a. All fields as defined for X.509 version 1 certificates must be present and non-null.
    b. X.509 certificates with version numbers greater than 3 are not supported.
    c. Version 3 certificates with critical extensions are not supported. Noncritical extensions are ignored.
    d. Subject and issuer names can contain only the following string types:
       - T61STRING- TAG 20
       - PRINTABLESTRING- TAG 19
       - IA5STRING- TAG 22
       - VISIBLESTRING- TAG 26
       - GENERALSTRING- TAG 27
       - BMPString-TAG 30 (ASCII Unicode only)
       - UTF8-TAG 12 (7 bit ASCII only)
    e. The length of the serial number plus the length of the issuer's name cannot exceed 245.
    f. No date validity check is performed on the certificate.
    g. No signature check is performed on the certificate.

    If the function_code indicates that an ACEE is to be created, or that a certificate is to be queried, the certificate must be a single DER encoded X.509 certificate.

    If the function_code indicates that a certificate is to be registered or deregistered, it must be in one of the following formats:
    a. A single DER encoded X.509 certificate.
    b. A Privacy Enhanced Mail (PEM) encoded X.509 certificate. If the input is in this format, only the Originator Certificate is used.
    c. One or more X.509 certificates contained within a PKCS #7 DER encoding. If the input is in this format, only the first certificate in the PKCS #7 encoding will be used.
    d. A Base64 encoded X.509 certificate as returned from a PKCS #10 certificate request. The data must include the string

       ```
       '-----BEGIN CERTIFICATE-----'
       ```

       immediately prior to the Base64 encoding, and the string

       ```
       '-----END CERTIFICATE-----'
       ```

       immediately following.

If transmitted from an ASCII system, PEM and Base64 encoded certificates must be translated from ASCII to EBCDIC before being passed to initACEE.

20. If the function_code indicates that a certificate is to be queried, the caller is expected to supply a 9 byte area for the RACF_userid parameter. If a user ID is associated with the certificate, initACEE updates this area with the length and value of the user ID.
21. If the function_code indicates that an ACEE is to be created or that a certificate is to be queried, and the certificate supplied by the caller is defined to RACF with a status of NOTRUST, initACEE will return a RACF return code 8 and a RACF reason code 40, indicating that no user ID is defined to use this certificate.
22. If the function_code and attributes indicate that an ACEE is to be created and an ENVR object is to be returned, then the ENVR_out parameter must point to a data structure for the ENVR object. The caller receives a parameter list error if the high order bit of a previous parameter indicates the end of the parameter list.
23. If the attributes indicate that an ENVR object is to be returned, it is the caller's responsibility to free the ENVR object storage. The caller should check the storage area length and address to determine if storage needs to be freed, not the initACEE return code. In some cases, an error may be encountered after creation of the ENVR object, resulting in a non-zero return code. The caller is still responsible for freeing the ENVR object in these cases.
24. If the function_code indicates that an ACEE is to be created, and the ENVR_in parameter points to an ENVR object data structure, then the length of the RACF_userid, Password, and Certificate parameters must all be 0. The caller receives a parameter list error if a RACF_userid, Password, or Certificate is supplied with the ENVR_in parameter.
25. If the function_code indicates that an ACEE is to be deleted or that managed ACEEs should be purged, and the ENVR_in or ENVR_out parameter is specified, then the caller receives a parameter list error.
26. When an ENVR object is supplied with the ENVR_in parameter and an ACEE creation is requested, the attribute bits that affect the ACEE (INTA_UNAUTH_CLIENT, INTA_AUTH_CLIENT, INTA_NO_TIMEOUT) and the application name (APPL_id) are ignored.
27. An OUSP can only be returned if the Attributes parameter also indicates that a USP should be created (INTA_OUSP_RET should only be on if INTA_USP is on). If an OUSP is requested without a USP, then the caller receives a parameter list error.
28. If the Attributes parameter indicates that an OUSP should be returned (INTA_OUSP_RET), then the Output_area parameter must be specified. If it is not, the caller receives a parameter list error.
29. When the INTA_NO_TIMEOUT bit and the INTA_MANAGED bit are set for the creation of a new managed ACEE, the ACEE is cached and does not expire after five minutes.
30. If the Attributes parameter indicates that a no timeout ACEE is requested (INTA_NO_TIMEOUT) and a managed ACEE with an expiration time is found in the cache that satisfies the request, the address of the managed ACEE is returned. The expiration time of the ACEE in the cache remains the same. After receiving a subsequent delete request, the ACEE may expire.
31. If the function indicates that a certificate is being queried, and the Attributes parameter indicates that an X500 name pair should be returned (INTA_X500_RET), then both the DIGTCERT and DIGTNMAP profiles will be checked for a user ID associated with the certificate. If the function indicates

that a certificate is being queried, and the X500 name pair has not been requested, then only the DIGTCERT profiles will be checked to determine if the certificate has been defined to RACF, and associated with a user ID.

32. If the Attributes parameter indicates that an X500 name pair should be returned (INTA_X500_RET), then the X500name parameter must be specified. If it is not, the caller receives a parameter list error.
33. If a certificate is being queried, and an X500 name pair has been requested, DIGTNMAP profiles may be used to determine the RACF user ID. If there are additional criteria associated with the DIGTNMAP profile, the APPL_id and Variable_list parameters, as well as the system-identifier of the system initACEE is running on, are used in determining the RACF user ID. If the certificate supplied for the query will later be used to create an ACEE, and the same user ID is expected to result, then the additional criteria must be the same for the query and create functions. In other words, the APPL_id and Variable_list parameter specifications must be the same, and the query and create must execute on the same system.
34. When the ENVR_in parameter is specified, the INTA_USP attribute is ignored. If the ENVR_in contains a USP, the resulting ACEE will also have a USP associated with it. The X500name parameter will also be ignored. If the ENVR_in contains an X500 name, the resulting ACEE will also have an X500 name associated with it. The information needed to create an OUSP is not available in an ENVR_object. If the INTA_RET_OUSP attribute is set indicating an OUSP should be returned, and an ENVR object is supplied with the ENVR_in parameter, the caller receives a parameter list error.
35. If the function_code indicates that an ACEE is to be created or that a certificate is to be queried and processing determine that the user ID to be used is to be extracted from the hostIdMappings certificate extension, InitACEE will ignore the extension under the following conditions:
    - The caller does not have READ authority (or greater) to any SERVAUTH class resource identified by the hostName(s) in the extension.
    - The user ID extracted has a length less than 1 or greater than 8.
    - For create only, the user ID is not a RACF defined user.
    - The definition of the hostIdMappings extension in ASN.1 syntax is:

```
id-ce-hostIdMappings OBJECT IDENTIFIER::= {1 3 18 0 2 18}

HostIdMappings::= SET OF HostIdMapping

HostIdMapping::= SEQUENCE{
   hostName        IMPLICIT[1] IA5String,
   subjectId         IMPLICIT[2] IA5 String,
   proofOfIdPossession IdProof OPTIONAL
 }
 IdProof::= SEQUENCE{
   secret        OCTET STRING,
   envryptionAlgorithm OBJECT IDENTIFIER
 }
```

36. If the function_code indicates that a certificate is to be registered, and one of the CERTAUTH certificates meets the following three conditions, the input certificate is treated as a certificate authority certificate to be registered as CERTAUTH:
    - The CERTAUTH certificate's public key matches that of the input certificate.
    - The CERTAUTH certificate's subject distinguished name matches that of the input certificate and

- The CERTAUTH certificate has a private key.

Otherwise, the input certificate is treated as an end-user certificate to be registered with the current user ID.

37. When a return code of 8 8 40 is received from an initACEE Create, other occurrances could be if the DIGTCERT, DIGTNMAP, or DIGTCRIT class has not been processed using SETROPTS RACLIST, or if SETROPTS RACLIST was used, but the class was not RACLIST REFRESHed after the certificate or mapping was added or altered. See *OS/390 SecureWay Security Server RACF Command Language Reference* for information on the RACDCERT and SETROPTS commands.

## Related Services

None

---

# initUSP (IRRSIU00): Initialize USP

## Function

The **initUSP** service verifies that the user is authorized to use OS/390 UNIX System Services and, if so, establishes security attributes for the calling process. The **initUSP** service also returns any OS/390 UNIX System Services limits that have been set on a user basis.

## Requirements

| | |
|---|---|
| **Authorization:** | Any PSW key in supervisor state |
| **Dispatchable unit mode:** | Task of OS/390 UNIX user |
| **Cross memory mode:** | PASN = HASN |
| **AMODE:** | 31 |
| **RMODE:** | Any |
| **ASC mode:** | Primary or AR mode |
| **Recovery mode:** | ESTAE. Caller cannot have an FRR active. |
| **Serialization:** | Enabled for interrupts |
| **Locks:** | No locks held |
| **Control parameters:** | The parameter list and the work area must be in the primary address space. ALETs must be passed for all parameters except the work area. The words containing the ALETs must be in the primary address space. |

## RACF Authorization

None

## Format

```
CALL IRRSIU00 (Work_area,
               ALET, SAF_return_code,
               ALET, RACF_return_code,
               ALET, RACF_reason_code,
               ALET, Output_area
              )
```

## Parameters

**Work_area**
The name of a 1024-byte work area for SAF and RACF usage. The work area must be in the primary address space and start on a word boundary.

**ALET**
The name of a word containing the ALET for the following parameter. Each parameter must have an ALET specified. Each ALET can be different. The words containing the ALETs must be in the primary address space.

**SAF_return_code**
The name of a fullword in which the SAF router returns the SAF return code.

**RACF_return_code**
The name of a fullword in which the service routine stores the return code.

**RACF_reason_code**
The name of a fullword in which the service routine stores the reason code.

**Output_area**
The name of a fullword in which the service routine stores the address of an area containing data about the user. IRRSIU00 uses the high-order bit of this fullword to determine if OS/390 UNIX requested default processing for failures that occur under certain circumstances (as described in Note 1). If the high-order bit is on, an initUSP that would normally fail because of missing information completes successfully and builds a default USP. With current default processing (described for the getUMAP and getGMAP callable services and in usage notes for this service) even without the high-order bit turned on under the circumstances described in Notes 4 and 5, an initUSP that would previously fail is now completed successfully.

For all successful initUSP requests, the output area is obtained in the primary address space and must be freed by the caller of initUSP. The following data is returned:
- TSO/E user ID
- OS/390 UNIX user identifier (UID) of user
- OS/390 UNIX group identifier (GID) of current group
- Home directory path name
- Initial program path name
- User limits (when OUSP version is greater than 0)

The actual format of the output area is mapped by macro IRRPOUSP.

See *OS/390 SecureWay Security Server RACF Data Areas* for the actual format of the output area.

## Return and Reason Codes

IRRSIU00 returns the following values in the reason and return code parameters:

| SAF Return Code | RACF Return Code | RACF Reason Code | Explanation |
|---|---|---|---|
| 0 | 0 | 0 | The service was successful. |
| 4 | 0 | 0 | RACF is not installed. |
| 8 | 8 | 12 | An internal error occurred during RACF processing. |
| 8 | 8 | 16 | Recovery could not be established. |

| SAF Return Code | RACF Return Code | RACF Reason Code | Explanation |
|---|---|---|---|
| 8 | 8 | 20 | The user's profile has no OMVS segment. |
| 8 | 8 | 24 | The OMVS segment in the user's profile has no UID. |
| 8 | 8 | 28 | The OMVS segment in the current group's profile has no GID. |

## Usage Notes

1. This service is intended only for use by OS/390 UNIX System Services servers and the MVS BCP. It can be directly invoked by an OS/390 UNIX System Services server. The high-order bit of the output_area is set on by the caller when initUSP is called to establish the security attributes for critical OS/390 UNIX address spaces such as the kernel. When the bit is on, initUSP builds a default OS/390 UNIX security environment in certain cases when it would normally fail. InitUSP sets a SAF return code of 0, RACF return code of 0, RACF reason code of 0, and builds a default USP for the following cases:
   - The user ID is not defined to RACF
   - There is no OMVS segment in the user's profile
   - There is no UID in the OMVS segment of the user's profile
   - There is no GID in the OMVS segment of the current group's profile

   The default USP returned to the caller (mapped by IRRPOUSP) contains a OS/390 UNIX user identifier (UID) and OS/390 UNIX group identifier (GID) of 0. The lengths for initial program and home directory path names is 0. If the user ID is defined to RACF, the user ID is returned as an TSO/E user ID. If the user ID is not defined to RACF, the TSO/E user ID is set to an asterisk in the returned USP.
2. The address space or task must have an ACEE when this service is called.
3. A RACF user can be connected to more than NGROUPS_MAX groups, but only up to the first NGROUPS_MAX groups will be associated with the process when the process is created.

   The first NGROUPS_MAX OS/390 UNIX groups to which a user is connected (as shown by a LISTUSER command) get associated with the process.
4. If no OMVS segment is found in the user's profile, the initUSP service checks the BPX.DEFAULT.USER profile in the FACILITY class. This profile may contain a user ID in its application data field that provides a default OMVS segment. If this profile is found and the FACILITY class is active, it is used to set the home, PROGRAM, and user limits for the user.
5. If no OMVS segment is found in the group profile of the user's current connect group, the initUSP service checks the BPX.DEFAULT.USER profile in the FACILITY class. This profile may contain a group name in its application data field that provides a default OMVS segment. If this profile is found and the FACILITY class is active, it is used to set the GID for the user.

   See *OS/390 SecureWay Security Server RACF Security Administrator's Guide* for information on APPLDATA in the FACILITY class.

## Related Services

deleteUSP

# makeFSP (IRRSMF00): Make IFSP

## Function

The **makeFSP** service builds an IFSP in the area provided by the caller.

## Requirements

| | |
|---|---|
| **Authorization:** | Any PSW key in supervisor state |
| **Dispatchable unit mode:** | Task of OS/390 UNIX user |
| **Cross memory mode:** | PASN = HASN or PASN not = HASN |
| **AMODE:** | 31 |
| **RMODE:** | Any |
| **ASC mode:** | Primary or AR mode |
| **Recovery mode:** | SETFRR |
| **Serialization:** | Enabled for interrupts |
| **Locks:** | No locks held |
| **Control parameters:** | The parameter list and the work area must be in the primary address space. ALETs must be passed for all parameters except the work area. The words containing the ALETs must be in the primary address space. |

## RACF Authorization

None

## Format

```
CALL IRRSMF00 (Work_area,
               ALET, SAF_return_code,
               ALET, RACF_return_code,
               ALET, RACF_reason_code,
               ALET, Mode,
               ALET, Output_FSP,
               ALET, Owning_directory_FSP,
               ALET, File_Identifier,
               ALET, CRED
              )
```

## Parameters

**Work_area**
The name of a 1024-byte work area for SAF and RACF usage. The work area must be in the primary address space.

**ALET**
The name of a word containing the ALET for the following parameter. Each parameter must have an ALET specified. Each ALET can be different. The words containing the ALETs must be in the primary address space.

**SAF_return_code**
The name of a fullword in which the SAF router returns the SAF return code.

**RACF_return_code**
The name of a fullword in which the service routine stores the return code.

**RACF_reason_code**
The name of a fullword in which the service routine stores the reason code.

**Mode**
The name of a word containing the mode values (the filetype, the permission bits, and the S_ISUID, S_ISGID, and S_ISVTX bits) to be set for the file.

See "File Type and File Mode Values" on page 4 for a definition of the security bits in the mode parameter.

**Output_FSP**
The name of a 64-byte area in which the new IFSP is built.

**Owning_directory_FSP**
The name of an area containing the IFSP for the owning directory. The owning OS/390 UNIX group identifier (GID) from this IFSP becomes the owning GID of the new file.

**File_Identifier**
The name of a 16-byte area containing a unique identifier of the file.

**CRED**
The name of the CRED structure for the current file system syscall. See *OS/390 SecureWay Security Server RACF Data Areas*.

## Return and Reason Codes

IRRSMF00 returns the following values in the reason and return code parameters:

| SAF Return Code | RACF Return Code | RACF Reason Code | Explanation |
|---|---|---|---|
| 0 | 0 | 0 | The service was successful. |
| 4 | 0 | 0 | RACF is not installed. |
| 8 | 8 | 12 | An internal error occurred during RACF processing. |
| 8 | 8 | 32 | CRED user type is not supported. |

## Usage Notes

1. This service is only intended for use by an OS/390 UNIX System Services file system and by OS/390 UNIX System Services servers. The service contains support for OS/390 UNIX System Services servers, but cannot be directly invoked by an OS/390 UNIX System Services server.
2. If the CRED user type is system, IRRSMF00 allows the operation, and sets the owning OS/390 UNIX user identifier (UID) to zero.
3. IRRSMF00 builds the IFSP in the output_FSP area provided by the caller. The caller must save the IFSP as part of the attributes for the object.
4. IRRSMF00 builds the IFSP with the S_ISUID and S_ISGID bits set to zero and the S_ISVTX bit set to the value in the mode byte.
5. An audit record is optionally written, depending on the audit options in effect for the system.

## Related Services

ck_access, R_umask

# makeISP (IRRSMI00): Make IISP

## Function

The **makeISP** service builds an IISP in the area provided by the caller.

## Requirements

| | |
|---|---|
| **Authorization:** | Any PSW key in supervisor state |
| **Dispatchable unit mode:** | Task of OS/390 UNIX user |
| **Cross memory mode:** | PASN = HASN or PASN not = HASN |
| **AMODE:** | 31 |
| **RMODE:** | Any |
| **ASC mode:** | Primary or AR mode |
| **Recovery mode:** | SETFRR |
| **Serialization:** | Enabled for interrupts |
| **Locks:** | No locks held |
| **Control parameters:** | The parameter list and the work area must be in the primary address space. ALETs must be passed for all parameters except the work area. The words containing the ALETs must be in the primary address space. |

## RACF Authorization

None

## Format

```
CALL IRRSMI00 (Work_area,
               ALET, SAF_return_code,
               ALET, RACF_return_code,
               ALET, RACF_reason_code,
               ALET, Mode_Permissions,
               ALET, Output_ISP,
               ALET, Output_IPCP,
               ALET, CREDIPC
              )
```

## Parameters

**Work_area**
The name of a 1024-byte work area for SAF and RACF usage. The work area must be in the primary address space.

**ALET**
The name of a word containing the ALET for the following parameter. Each parameter must have an ALET specified. Each ALET can be different. The words containing the ALETs must be in the primary address space.

**SAF_return_code**
The name of a fullword in which the SAF router returns the SAF return code.

**RACF_return_code**
The name of a fullword in which the service routine stores the return code.

**RACF_reason_code**
The name of a fullword in which the service routine stores the reason code.

**Mode_Permissions**
The name of a word containing the mode permission flags to be set for this IPC key. The following is a list of defined permission bits mapped by BPXYMODE:

**S_IRUSR**
Permits the process that owns the IPC member to read it.

**S_IWUSR**
Permits the process that owns the IPC member to alter it.

**S_IRGRP**
Permits the group associated with the IPC member to read it.

**S_IWGRP**
Permits the group associated with the IPC member to alter it.

**S_IROTH**
Permits others to read the IPC member.

**S_IWOTH**
Permits others to alter the IPC member.

Alter and write have the same meaning for access checks. Alter applies to semaphores and write applies to message queueing and shared memory segments.

**Output_ISP**
The name of a 64-byte area in which the new IISP is built. The name is set by the kernel. See *OS/390 SecureWay Security Server RACF Data Areas*.

**Output_IPCP**
The name of a 20-byte area in which the new IPCP is built. The name is set by the kernel.

**CREDIPC**
The name of the CREI structure for the current IPC system callable service. The CREI contains the IPC identifier and IPC key. See *OS/390 SecureWay Security Server RACF Data Areas*.

## Return and reason codes

IRRSMI00 may return the following values in the reason and return code parameters:

| SAF return code | RACF return code | RACF reason code | Explanation |
|---|---|---|---|
| 0 | 0 | 0 | The service was successful. |
| 4 | 0 | 0 | RACF is not installed. |
| 8 | 8 | 12 | An internal error occurred during RACF processing. |
| 8 | 8 | 32 | The CREI user type is not supported. |

## Usage Notes

1. This service is only intended for use by the MVS BCP.
2. The CREI user type must be local (that is, 1).

3. IRRSMI00 builds the IISP in the output_ISP area and the output_IPCP areas provided by the caller. The caller must save the IISP as part of the attributes for the key.
4. The IPCP ALET and address are retrieved from the parameters and set into the output_ISP by RACF.
5. The effective OS/390 UNIX user identifier (UID) and OS/390 UNIX group identifier (GID) are retrieved from the USP and set into the owner and creator fields of the output_IPCP by RACF.
6. The mode is retrieved from the parameters and set into the output_IPCP by RACF.
7. The IPC Key and IPC ID are retrieved from the CREI and set into the output_ISP by RACF.
8. An audit record is optionally written, depending on the audit options in effect for the system.
9. This service uses task level support when OS/390 UNIX has indicated in the task's ACEE that this is a task level process.

## Related services

ck_IPC_access, R_IPC_ctl

---

# make_root_FSP (IRRSMR00): Make Root IFSP

## Function

The **make_root_FSP** service initializes an IFSP for the root directory of a new file system being initialized in a hierarchical file system (HFS) data set.

## Requirements

| | |
|---|---|
| **Authorization:** | Any PSW key in supervisor state |
| **Dispatchable unit mode:** | Task |
| **Cross memory mode:** | PASN = HASN |
| **AMODE:** | 31 |
| **RMODE:** | Any |
| **ASC mode:** | Primary or AR mode |
| **Recovery mode:** | ESTAE. Caller cannot have an FRR active. |
| **Serialization:** | Enabled for interrupts |
| **Locks:** | No locks held |
| **Control parameters:** | The parameter list and the work area must be in the primary address space. ALETs must be passed for all parameters except the work area. The words containing the ALETs must be in the primary address space. |

## RACF Authorization

None

## Format

```
CALL IRRSMR00 (Work_area,
               ALET, SAF_return_code,
               ALET, RACF_return_code,
               ALET, RACF_reason_code,
               ALET, Mode,
               ALET, Output_FSP,
               ALET, File_Identifier,
               ALET, Data_set_name
              )
```

## Parameters

**Work_area**
The name of a 1024-byte work area for SAF and RACF usage. The work area must be in the primary address space.

**ALET**
The name of a word containing the ALET for the following parameter. Each parameter must have an ALET specified. Each ALET can be different. The words containing the ALETs must be in the primary address space.

**SAF_return_code**
The name of a fullword in which the SAF router returns the SAF return code.

**RACF_return_code**
The name of a fullword in which the service routine stores the return code.

**RACF_reason_code**
The name of a fullword in which the service routine stores the reason code.

**Mode**
The name of a word containing the mode value (the file type, the permission bits, and the S_ISUID, S_ISGID, and S_ISVTX bits) to be set for the file.

See "File Type and File Mode Values" on page 4 for a definition of the security bits in the mode parameter.

**Output_FSP**
The name of a 64-byte area in which the new IFSP is built.

**File_Identifier**
The name of a 16-byte area containing a unique identifier of the root directory.

**Data_set_name**
The name of an area containing the data set name of the HFS data set being created. This is a 44-byte area padded with blanks.

## Return and Reason Codes

IRRSMR00 returns the following values in the reason and return code parameters:

| SAF Return Code | RACF Return Code | RACF Reason Code | Explanation |
|---|---|---|---|
| 0 | 0 | 0 | The service was successful. |
| 4 | 0 | 0 | RACF is not installed. |
| 8 | 8 | 12 | An internal error occurred during RACF processing. |
| 8 | 8 | 16 | Recovery could not be established. |

## Usage Notes

1. This service is only intended for use by DFSMS/MVS, during allocation of an HFS data set, and by OS/390 UNIX System Services servers. The service contains support for OS/390 UNIX System Services servers, but can not be directly invoked by an OS/390 UNIX System Services server.
2. IRRSMR00 may be called from a non-OS/390 UNIX address space.
3. These are the default attributes set for the root directory:
   - The file's owner OS/390 UNIX user identifier (UID) and OS/390 UNIX group identifier (GID) are initialized as follows:
     - If the caller is an OS/390 UNIX process:
       owner UID = effective UID of the process
       owner GID = effective GID of the process

       **Note:** This differs from **makeFSP** because there is no owning directory to propagate the GID from.
     - If the caller is not an OS/390 UNIX process but is defined to RACF as an OS/390 UNIX user:
       owner UID = UID from the user's profile
       owner GID = GID from the group profile for the user's current group

       If the group has no GID, the owner GID is set to 0.
     - If the caller is not an OS/390 UNIX process and is not defined to RACF as an OS/390 UNIX user:
       owning UID = 0
       owning GID = 0

       If the UID or GID is set to 0, a superuser should change the fields to valid values using **chown** after the file system is mounted.
   - The permission bits are set from the input mode parameter.
   - The S_ISUID and S_ISGID are set to 0 and the S_ISVTX bit is set to the value in the input mode parameter.
   - The user audit options are set to audit access failures for all types of access.
   - The auditor audit options are set to no auditing.
4. IRRSMR00 builds the IFSP in the output_FSP area provided by the caller. The caller must save the IFSP as part of the attributes for the object.

## Related Services

makeFSP

---

# query_file_security_options (IRRSQF00): Query File Security Options

## Function

The **query_file_security_options** service returns the value of the requested file system option. The only file system option that can be queried is _POSIX_CHOWN_RESTRICTED.

This service returns a value of 0 if _POSIX_CHOWN_RESTRICTED is in effect and a value of -1 if _POSIX_CHOWN_RESTRICTED is not in effect.

## Requirements

**Authorization:** Any PSW key in supervisor state

| | |
|---|---|
| **Dispatchable unit mode:** | Task of OS/390 UNIX user |
| **Cross memory mode:** | PASN = HASN or PASN not = HASN |
| **AMODE:** | 31 |
| **RMODE:** | Any |
| **ASC mode:** | Primary or AR mode |
| **Recovery mode:** | None |
| **Serialization:** | Enabled for interrupts |
| **Locks:** | No locks held |
| **Control parameters:** | The parameter list and the work area must be in the primary address space. ALETs must be passed for all parameters except the work area. The words containing the ALETs must be in the primary address space. |

## RACF Authorization

1. **The default setting on OS/390 is that _POSIX_CHOWN_RESTRICTED is in effect.**

   If all of the following are true, then _POSIX_CHOWN_RESTRICTED is not in effect:

   - The UNIXPRIV class is active
   - The UNIXPRIV class has been processed using SETROPTS RACLIST
   - The following discrete profile exists in the UNIXPRIV class: CHOWN.UNRESTRICTED.

## Format

```
CALL IRRSQF00 (Work_area,
               ALET, SAF_return_code,
               ALET, RACF_return_code,
               ALET, RACF_reason_code,
               ALET, Option_code,
               ALET, Output_value
              )
```

## Parameters

**Work_area**
The name of a 1024-byte work area for SAF and RACF usage. The work area must be in the primary address space.

**ALET**
The name of a word containing the ALET for the following parameter. Each parameter must have an ALET specified. Each ALET can be different. The words containing the ALETs must be in the primary address space.

**SAF_return_code**
The name of a fullword in which the SAF router returns the SAF return code.

**RACF_return_code**
The name of a fullword in which the service routine stores the return code.

**RACF_reason_code**
The name of a fullword in which the service routine stores the reason code.

**Option_code**
The name of a word containing a code identifying the requested option. The code value 1 identifies the _POSIX_CHOWN_RESTRICTED option. All other code values are reserved.

**Output_value**
The name of a word in which the value of the requested option is returned. The values for _POSIX_CHOWN_RESTRICTED are:

**0** _POSIX_CHOWN_RESTRICTED is in effect

**-1** _POSIX_CHOWN_RESTRICTED is not in effect

## Return and Reason Codes

IRRSQF00 returns the following values in the reason and return code parameters:

| SAF Return Code | RACF Return Code | RACF Reason Code | Explanation |
|---|---|---|---|
| 0 | 0 | 0 | The service was successful. |
| 4 | 0 | 0 | RACF is not installed. |
| 8 | 8 | 4 | The option code is not supported. |

## Usage Note

1. This service is intended only for use by an OS/390 UNIX System Services file system.

## Related Services

None

# query_system_security_options (IRRSQS00): Query System Security Options

## Function

The **query_system_security_options** service returns the value of the requested system options. The only supported options are NGROUPS_MAX and _POSIX_SAVED_IDS.

## Requirements

**Authorization:** Any PSW key in supervisor state
**Dispatchable unit mode:** Task of OS/390 UNIX user
**Cross memory mode:** PASN = HASN or PASN not = HASN
**AMODE:** 31
**RMODE:** Any
**ASC mode:** Primary or AR mode
**Recovery mode:** None
**Serialization:** Enabled for interrupts
**Locks:** No locks held
**Control parameters:** The parameter list and the work area must be in the primary address space. ALETs must be passed for all parameters except the work area. The words containing the ALETs must be in the primary address space.

## RACF Authorization

1. RACF returns the following values for the requested system options:
   - NGROUPS_MAX: 300
   - _POSIX_SAVED_IDS: 0

## Format

```
CALL IRRSQS00 (Work_area,
               ALET, SAF_return_code,
               ALET, RACF_return_code,
               ALET, RACF_reason_code,
               ALET, Option_code,
               ALET, Output_value
              )
```

## Parameters

**Work_area**
The name of a 1024-byte work area for SAF and RACF usage. The work area must be in the primary address space.

**ALET**
The name of a word containing the ALET for the following parameter. Each parameter must have an ALET specified. Each ALET can be different. The words containing the ALETs must be in the primary address space.

**SAF_return_code**
The name of a fullword in which the SAF router returns the SAF return code.

**RACF_return_code**
The name of a fullword in which the service routine stores the return code.

**RACF_reason_code**
The name of a fullword in which the service routine stores the reason code.

**Option_code**
The name of a word containing a code identifying the requested option. The supported values are:

**1** NGROUPS_MAX

**2** _POSIX_SAVED_IDS

All other values are reserved.

**Output_value**
The name of a word in which the value of the requested option is returned.

- The values for _POSIX_SAVED_IDS are:

  **0** _POSIX_SAVED_IDS is in effect.

  **-1** _POSIX_SAVED_IDS is not in effect.
- The value for NGROUPS_MAX is the maximum number of supplemental groups supported.

## Return and Reason Codes

IRRSQS00 returns the following values in the reason and return code parameters:

| SAF Return Code | RACF Return Code | RACF Reason Code | Explanation |
|---|---|---|---|
| 0 | 0 | 0 | The service was successful. |
| 4 | 0 | 0 | RACF is not installed. |
| 8 | 8 | 4 | The option code is not supported. |

## Usage Note

- This service is intended only for use by the MVS BCP.

## Related Services

None

# R_admin (IRRSEQ00): RACF Administration API

## Function

The **R_admin** service enables applications to construct a function code/data field name parameter list to manage RACF profiles within the RACF database. Alternately, a RACF TSO administrative command image can be passed to manage RACF user profile information. Output, if any, which resulted from RACF's processing of the profile admin request is returned to the caller in virtual storage. This callable service does **not** support all RACF command functions. For a list of the commands that are **not** executed through this service, see the Usage Notes section.

The exact format (spacing and order) of the data in the command output or messages does not form a programming interface. No programs should depend on the exact format of this data.

## Requirements

**Authorization:** Any PSW key in supervisor or problem state
**Dispatchable unit mode:** Any task
**Cross memory mode:** PASN = HASN = SASN
**AMODE:** 31
**RMODE:** Any
**ASC mode:** Primary only
**Recovery mode:** Recovery must be provided by caller
**Serialization:** Enabled for interrupts
**Locks:** No locks held
**Control parameters:** The parameter list and the work area must be in the primary address space.

## RACF Authorization

For the function codes which cause a RACF command to execute in the RACF address space, the command will run under the authority of a RACF user ID. The RACF user ID can come from one of a number of sources, and is searched for in the following order:

- The RACF_userID parameter
- The ACEE_ptr parameter
- The user ID associated with the current task control block (TCB)
- The user ID associated with the current address space (ASXB)

For problem state callers only, READ authority to the resource IRR.RADMIN.*(command-name)* in the FACILITY class is required to execute the RACF TSO command *command-name* using R_Admin. This is in addition to any authority checks done by the command itself. The resource must be defined using the full command name even if the abbreviated version of the command name is used with R_Admin. (e.f., lu joeuser would require READ authority to IRR.RADMIN.LISTUSER. Generic profiles can be used.

## Format

```
CALL IRRSEQ00 (Work_area,
               ALET, SAF_return_code,
               ALET, RACF_return_code,
               ALET, RACF_reason_code,
               Function_code,
               Parm_list,
               RACF_userID,
               ACEE_ptr,
               Out_message_subpool
               Out_message_strings
              )
```

## Parameters

**Work_area**
The name of a 1024-byte work area for SAF and RACF usage. The work area must be in the primary address space.

**ALET**
The name of a word containing the ALET for the following parameter. Each parameter must have an ALET specified. Each ALET can be different. The word containing the ALETs must be in the primary address space.

**SAF_return_code**
The name of a fullword in which the SAF router returns the SAF return code. These return codes are found in Table 21 on page 65.

**RACF_return_code**
The name of a fullword in which the service routine stores the return code. These return codes are found in Table 21 on page 65.

**RACF_reason_code**
The name of a fullword in which the service routine stores the reason code. These reason codes are found in Table 21 on page 65.

**Function_code**
The name of a one-byte field that specifies the function (administration request) that RACF is to perform. The function code may have one of the values found in Table 17 on page 63.

**Parm_list**
The name of a variable marking the start of the input parameter list. The mapping macro IRRPCOMP contains a definition of the parameter list for each

of the values of function_code. To find parameter list mappings for the values of function_code, see Table 18 on page 64.

**ACEE_ptr**
The name of a fullword containing the address of the ACEE of the user under whose identity the RACF administrative request runs. The user ID is extracted from the ACEEUSER field. The ACEE itself is not used for subsequent authority checking for the request. If the caller does not specify an ACEE, this area must contain binary zeros. If both an ACEE and a user ID are passed into this service, the user ID is used.

**Out_message_subpool**
The name of a one-byte field that specifies the subpool used to obtain storage for output messages that are returned.

**Out_message_strings**
The name of a fullword in which the service routine stores the address of output data, if applicable. It is the responsibility of the caller to free the output storage.

For SETROPTS functions (ADMN_XTR_SETR and ADMN_UNL_SETR), see Table 63 on page 97 for the output mapping.

For the remaining functions, see Table 19 on page 64 for the mapping of the output message block returned by this service. For the format of each message entry, see Table 20 on page 65. The message entry is a repeating data structure. In it, the entry length and message text repeats if more than one message is present. See also the IRRPCOMP data area in *OS/390 SecureWay Security Server RACF Data Areas*.

## Function Code Values

Table 17 shows the function code values in the mapping macro IRRPCOMP.

*Table 17. Function Code Values in Mapping Macro IRRPCOMP*

| Function Code | Value | Description |
|---|---|---|
| ADMN_ADD_USER | X'01' | Add a user to the RACF database |
| ADMN_DEL_USER | X'02' | Delete a user from the RACF database |
| ADMN_ALT_USER | X'03' | Alter a user's RACF database profile |
| ADMN_LST_USER | X'04' | List the contents of a user's RACF database profile |
| ADMN_RUN_COMD | X'05' | Run a RACF command image |
| ADMN_ADD_GROUP | X'06' | Add a group to the RACF database |
| ADMN_DEL_GROUP | X'07' | Delete a group from the RACF database |
| ADMN_ALT_GROUP | X'08' | Alter a group's RACF database profile |
| ADMN_LST_GROUP | X'09' | List a group's RACF database profile |
| ADMN_CONNECT | X'0A' | Connect a single user to a RACF group |
| ADMN_REMOVE | X'0B' | Remove a single user from a RACF group |
| ADMN_ADD_GENRES | X'0C' | Add a general resource profile to the RACF database |
| ADMN_DEL_GENRES | X'0D' | Delete a general resource profile from the RACF database |
| ADMN_ALT_GENRES | X'0E' | Alter a general resource's RACF database profile |
| ADMN_LST_GENRES | X'0F' | List a general resource's RACF database profile |

*Table 17. Function Code Values in Mapping Macro IRRPCOMP (continued)*

| Function Code | Value | Description |
|---|---|---|
| ADMN_ADD_DS | X'10' | Add a data set profile to the RACF database |
| ADMN_DEL_DS | X'11' | Delete a data set profile from the RACF database |
| ADMN_ALT_DS | X'12' | Alter a data set's RACF database profile |
| ADMN_LST_DS | X'13' | List a data set's RACF database profile |
| ADMN_PERMIT | X'14' | Permit a user or group to a RACF profile |
| ADMN_ALT_SETR | X'15' | Alter SETROPTS information |
| ADMN_XTR_SETR | X'16' | Extract SETROPTS information in R_admin format |
| ADMN_UNL_SETR | X'17' | Extract SETROPTS information in SMF data unload format |

## Parameter List Mappings by Function Code

Table 18 shows where to find the parameter list mappings for the values of function_code.

*Table 18. Parameter List Mappings for Function_Code Values*

| For function_code(s) | See |
|---|---|
| ADMN_ADD_USER, ADMN_DEL_USER, ADMN_ALT_USER, ADMN_LST_USER | "User Administration" on page 68 |
| ADMN_RUN_COMD | "Running RACF Commands" on page 68 |
| ADMN_ADD_GROUP, ADMN_DEL_GROUP, ADMN_ALT_GROUP, ADMN_LST_GROUP | "Group Administration" on page 77 |
| ADMN_CONNECT ADMN_REMOVE | "Group Connection Administration" on page 80 |
| ADMN_ADD_GENRES, ADMN_DEL_GENRES, ADMN_ALT_GENRES, ADMN_LST_GENRES, ADMN_ADD_DS, ADMN_DEL_DS, ADMN_ALT_DS, ADMN_LST_DS, ADMN_PERMIT | "Resource Profile Administration" on page 82 |
| ADMN_ALT_SETR | "Parameter List Format for SETROPTS Administration" on page 91 |
| ADMN_XTR_SETR | Input parameter list is ignored |
| ADMN_UNL_SETR | Input parameter list is ignored |

## Output Message Block Mapping

*Table 19. Mapping of Output Message Block*

| Offset | Length | Description |
|---|---|---|
| 0 | 4 | Next ADMIN output messages block, or zero if no additional blocks follow |
| 4 | 4 | Eye catcher to aid in virtual storage dumps 'RMSG' |
| 8 | 1 | Storage subpool in which the block was obtained |
| 9 | 3 | Total block length |
| 12 | 4 | Offset to the first byte after the last message; This offset value is related to the first message. |

*Table 19. Mapping of Output Message Block (continued)*

| Offset | Length | Description |
|---|---|---|
| 16 | 1 | Start of the first message |

*Table 20. Format of Each Message Entry*

| Offset | Length | Description |
|---|---|---|
| 0 | 2 | Length of this message text entry. |
| 2 | * | Variable message text. |

The actual format of the output area is mapped by macro IRRPCOMP.

See *OS/390 SecureWay Security Server RACF Data Areas* for the actual format of the output area.

## Return and Reason Codes

IRRSEQ00 returns the following values in the reason and return code parameters:

*Table 21. Return and Reason Codes*

| SAF Return Code | RACF Return Code | RACF Reason Code | Explanation |
|---|---|---|---|
| 0 | 0 | 0 | The service was successful. Output from the RACF command may be present. The Out_message_strings parameter should be interrogated by caller. |
| 4 | 0 | 0 | RACF is not installed. |
| 8 | 8 | 0 | Incorrect function code |
| 8 | 8 | 4 | Input parameter list error |
| 8 | 8 | 8 | Invalid data in the input parameter list may cause the program to abend. Some fields in the parameter list must be coded with the defined length and data format. The profile segment name, for example must not be longer than 8 bytes. If more than 8 bytes are passed, unpredictable results may occur and the user gets a return code of 8 8 8 with a possible program abend.<br><br>If the program ABENDs before GTF trace records are created, then you do not have GTF trace records to use for debugging. Check the input parameter list for errors. |
| 8 | 8 | 12 | Recovery environment could not be established. |

*Table 21. Return and Reason Codes (continued)*

| SAF Return Code | RACF Return Code | RACF Reason Code | Explanation |
|---|---|---|---|
| 8 | 8 | 16 | Invalid request specified. For ADMN_RUN_COMD, the input command image represented an incorrect or unsupported command. For all other functions, the input parameter list contained an incorrect segment name, field name, or flag byte, or incorrectly specified field data. The function-specific parameter list header contains an offset to the segment or field entry in error, relative to the start of the parameter list. |
| 8 | 8 | 20 | Function not supported for problem state caller. |
| 8 | 8 | 24 | Problem state caller not authorized to issue command. |
| 8 | 12 | *IEFSSREQ return code* | Unable to invoke the RACF subsystem. Reason code field contains a return code from the IEFSSREQ macro invocation. See *OS/390 MVS Using the Subsystem Interface* for information on possible return codes from IEFSSREQ. |
| 8 | 16 | *RACF command return code* | RACF request failed. Reason code field contains the return code from the RACF request. See *OS/390 SecureWay Security Server RACF Command Language Reference* for information on possible return codes from the RACF commands. In addition, diagnostic output resulting from the RACF command may be present. The Out_message_strings parameter should be interrogated by the caller. |

**Note:** Return and reason codes are shown in decimal.

## Usage Notes

1. You must link edit the IRRSEQ00 callable service stub into your application code to resolve the entry point address at run time.
2. For the Out_message_subpool parameter, select a subpool carefully. OS/390 makes certain assumptions about subpool usage and characteristics. Using subpool 0 or 250 or any subpool documented in *OS/390 Application Development Guide* as having a storage key of USER (for example, 227-231 and 241) may give unpredictable results.
3. The RACF subsystem must be active to use this service.
4. The caller of this service must free the message output blocks returned by this service.
5. All requests are processed synchronously. Control is not returned to the caller until RACF has processed the administration request and output, if any has been returned to the caller.
6. For the ADMN_RUN_COMD function code, the following RACF commands are not supported through this interface:

- BLKUPD
- RACLINK
- RVARY
- RACF operator commands (DISPLAY, RESTART, SET, SIGNOFF, STOP, and TARGET)

RACF TSO administrative commands may not be directed to other RACF remote sharing facility (RRSF) nodes. The command image passed by the caller cannot contain the keywords AT or ONLYAT. These keywords cause the command to fail with SAF return code 8, RACF return code 16, RACF reason code 8.

These messages are returned as command output:

IRRV013I subsystem-name SUBSYSTEM racf-command COMMAND FROM THE IRRSEQ00 CALLABLE SERVICE WAS NOT PROCESSED.

IRRV014I subsystem-name SUBSYSTEM AT() OR ONLYAT() KEYWORDS MAY NOT BE SPECIFIED WITH COMMANDS FROM THE IRRSEQ00 CALLABLE SERVICE.

7. The command text must be left-justified within the input buffer
8. If a RACF command runs and does not return any output, the Out_message_strings parameter will be zero.
9. The parameter list passed to this service is a variable-length (VL) parameter list. The high-order bit of the last field (address of Out_message_strings) must be set to mark the end of the parameter list.
10. All field data must be supplied in character format. For information about the contents of the field data, refer to *OS/390 SecureWay Security Server RACF Command Language Reference* for the appropriate command keyword as indicated in the following tables. For example, looking at Table 28 on page 72 to find details on the content of the HLDCLASS field, see the ADDUSER/ALTUSER documentation for the HOLDCLASS keyword of the TSO segment.

    Additionally, RACF has a restriction of no more than 255 operands affecting a single nonbase segment (such as the TSO segment in a user profile, or the TME segment in a general resource profile) on a single command. Since the R_admin callable service generates a RACF command, this restriction applies to the number of field operands affecting nonbase segments. See the ″RACF command restriction for nonbase segments in RACF profiles″ section in the Command Language Reference manual for specifics on that restriction.
11. For the list functions, the output for each segment is returned in the order in which the segments were supplied, with the exception that the BASE segment information is always returned first.
12. Any update to the RACF database caused by this service is subject to automatic direction and password synchronization as implemented by the installation.
13. The amount of output returned from any command run by this service is subject to the limits established for RRSF.
14. The following errors result in a ″input parameter list error″ being returned to the caller:
    - VL bit not set

- An incorrectly specified ADMN_USRADM_USER_LEN, ADMN_GRPADM_LEN, or ADMN_RESADM_CLAS_LEN (must be from 1-8, inclusively)
- An incorrectly specified length for the RACF user ID parameter (must be from 0 to 8, inclusively)
- Setting ADMN_USRADM_SEGM_NUM=0 on any of the list functions
- Omitting the PROFILE field on any of the general resource, data set (except list) or permit function codes

15. The only supported function for problem state callers is ADMIN_RUN_COMD. The RACF_userid and ACEE_ptr parameters are ignored for problem state callers.

## Related Services

None

## Parameter List Formats

The following information describes parameter list formats for running RACF commands and for all administration.

### Running RACF Commands

For the ADMN_RUN_COMD (execute a RACF command) function code, the mapping associated with the function-specific parameter list is mapped as follows:

*Table 22. Parameter List Format for Running a Command*

| Offset | Length | Description |
|---|---|---|
| 0 | 2 | Length of the RACF command string. Note, the length must not exceed 4096 characters. |
| 2 | * | Syntactically correct RACF TSO administration command string. |

### User Administration

For the ADMN_ADD_USER, ADMN_DEL_USR, ADMN_ALT_USER, and ADMN_LST_USER function codes, the mapping associated with the function specific parameter list is mapped as in Table 23.

*Table 23. Parameter List Format for User Administration*

| Offset | Length | Description |
|---|---|---|
| 0 | 1 | Length of the user ID |
| 1 | 8 | Uppercase RACF user ID |
| 9 | 1 | Reserved |
| 10 | 2 | Output offset to the segment or field entry in error in relationship to the start of the Parm_list. Only applies to ADMN_ADD_USER and ADMN_ALT_USER requests when an ″invalid request″ error is returned to the caller. |
| 12 | 2 | Number of RACF profile segments |
| 14 | * | Start of first segment entry |

**Note:** For ADMN_DEL_USER, no segment data is expected. The number of segments should be zero. If non-zero, any segment data present is ignored.

Each segment entry is mapped as in Table 24.

*Table 24. Segment Entry Mapping*

| Offset | Length | Description |
|---|---|---|
| 0 | 8 | Profile segment name (left-justified, uppercase, and padded with blanks). For ADMN_LST_USER, any user profile segment as defined in the RACF database templates is accepted. For ADMN_ADD_USER and ADMN_ALT_USER, the following segments are supported: BASE, CICS, DCE, DFP, KERB, LANGUAGE, LNOTES, NDS, NETVIEW, OMVS, OPERPARM, OVM, TSO and WORKATTR. |
| 8 | 1 | Flag byte. Use Y to create or alter the segment. Use N to delete the segment. |
| 9 | 2 | Number of fields to update within a segment |
| 11 | * | First field for segment |

For ADMN_LST_USER, the flag byte is ignored.

For ADMN_ADD_USER, the flag byte indicates whether the segment should be created (Y) or should no longer exist (N). Because the BASE segment cannot be deleted, the flag byte for BASE is ignored. In addition, the flag byte of 'I' may be specified for any segment name. When the callable service finds the 'I' flag byte, it ignores the segment.

For ADMN_LST_USER, ADMN_ADD_USER, ADMN_ALT_USER with the flag byte N, no field data is expected. The number of fields should be zero. If non-zero, any field data present is ignored.

Each field entry is mapped as in Table 25.

*Table 25. Field Entry Mapping*

| Offset | Length | Description |
|---|---|---|
| 0 | 8 | Field name as defined in the tables that follow. All field names must be left-justified, entered in all uppercase, and padded with blanks. |
| 8 | 1 | Field-specific flag byte |
| 9 | 2 | Length of field data |
| 11 | * | Field data |

In all of the tables that describe ADDUSER and ALTUSER, the following rules for ADMN_ADD_USER and ADMN_ALT_USER apply:

- For boolean fields, the flag byte value of 'Y' or 'N' determines the flag setting. No field data is allowed for the boolean fields. Coding field data results in an "Invalid Request" error being returned to the caller.
- For text fields, the flag byte 'Y' indicates the field should be created (or altered) with the specified field data. The flag byte 'N' indicates the field should be deleted (or not created). For flag byte 'N', any field data specified is ignored.
- For repeating text fields, a flag byte of 'A' indicates to add the specified field data to the existing data (if any). Field data must be specified. If field data is not

specified, an "Invalid Request" error is returned to the caller. The flag byte 'D' indicates to delete the specified data (if any).

In addition to the flag bytes specified in the following tables, the flag byte of 'I' may be specified for any segment name. When the callable service finds the 'I' flag byte, it ignores the segment.

The following tables define the field keywords and their usage. All field names relate directly to the ADDUSER and ALTUSER keywords. See *OS/390 SecureWay Security Server RACF Command Language Reference* for questions pertaining to field usage and data.

*Table 26. BASE Segment Fields*

| Field Name | Flag Byte Values | ADDUSER/ALTUSER Keyword Reference | Allowed on Add Requests | Allowed on Alter Requests |
|---|---|---|---|---|
| OWNER | 'Y' | OWNER(xx) | Yes | Yes |
| ADSP (Boolean) | 'Y' | ADSP | Yes | Yes |
| | 'N' | NOADSP | Yes | Yes |
| SPECIAL (Boolean) | 'Y' | SPECIAL | Yes | Yes |
| | 'N' | NOSPECIAL | Yes | Yes |
| OPER (Boolean) | 'Y' | OPERATIONS | Yes | Yes |
| | 'N' | NOOPERATIONS | Yes | Yes |
| GRPACC (Boolean) | 'Y' | GRPACC | Yes | Yes |
| | 'N' | NOGRPACC | Yes | Yes |
| PASSWORD | 'Y' | PASSWORD (xx) | Yes | Yes |
| | 'N' | NOPASSWORD | Yes | Yes |
| EXPIRED (Boolean) | 'Y' | EXPIRED | No | Yes |
| | 'N' | NOEXPIRED | No | Yes |
| NAME | 'Y' | NAME (xx) | Yes | Yes |
| | 'N' | NAME | No | Yes |
| DFLTGRP | 'Y' | DFLTGRP (xx) | Yes | Yes |
| GROUP | 'Y' | GROUP (xx) | No | Yes |
| AUTH | 'Y' | AUTHORITY (xx) | Yes | Yes |
| UACC | 'Y' | UACC (xx) | Yes | Yes |
| MODEL | 'Y' | MODEL (xx) | Yes | Yes |
| | 'N' | NOMODEL | No | Yes |
| DATA | 'Y' | DATA (xx) | Yes | Yes |
| | 'N' | NODATA | No | Yes |
| UAUDIT (Boolean) | 'Y' | UAUDIT | No | Yes |
| | 'N' | NOUAUDIT | No | Yes |
| AUDITOR (Boolean) | 'Y' | AUDITOR | Yes | Yes |
| | 'N' | NOAUDITOR | Yes | Yes |
| OIDCARD (Boolean) | 'Y' | OIDCARD | No | No |
| | 'N' | NOOIDCARD | Yes | Yes |

*Table 26. BASE Segment Fields (continued)*

| Field Name | Flag Byte Values | ADDUSER/ALTUSER Keyword Reference | Allowed on Add Requests | Allowed on Alter Requests |
|---|---|---|---|---|
| SECLEVEL | 'Y' | SECLEVEL (xx) | Yes | Yes |
| | 'N' | NOSECLEVEL | No | Yes |
| SECLABEL | 'Y' | SECLABEL (xx) | Yes | Yes |
| | 'N' | NOSECLABEL | No | Yes |
| CATEGORY | 'Y' | ADDCATEGORY(xx ...) | Yes | No |
| | 'A' | ADDCATEGORY(xx ...) | No | Yes |
| | 'D' | DELCATEGORY(xx ...) | No | Yes |
| NOTE: To remove unknown categories from the profile, specify the 'D' flag and a field length of zero. | | | | |
| REVOKE | 'Y' | REVOKE(xx) | No | Yes |
| RESUME | 'Y' | RESUME(xx) | No | Yes |
| WHENDAYS | 'Y' | WHEN(DAYS (xx)) | Yes | Yes |
| WHENTIME | 'Y' | WHEN(TIME (xx)) | Yes | Yes |
| CLAUTH | 'Y' | CLAUTH(xx...) | Yes | No |
| | 'A' | CLAUTH(xx ...) | No | Yes |
| | 'D' | NOCLAUTH(xx ...) | No | Yes |
| | 'N' | NOCLAUTH | Yes | No |
| REST | 'Y' | RESTRICTED | Yes | Yes |
| | 'N' | NORESTRICTED | Yes | Yes |

*Table 27. OMVS Segment Fields*

| Field Name | Flag Byte Values | ADDUSER/ALTUSER Keyword Reference | Allowed on Add Requests | Allowed on Alter Requests |
|---|---|---|---|---|
| OS/390 UNIX user identifier (UID) | 'Y' | OMVS( UID (xx)) | Yes | Yes |
| | 'N' | OMVS( (NOUID) | No | Yes |
| HOME | 'Y' | OMVS(HOME (xx)) | Yes | Yes |
| | 'N' | OMVS(NOHOME ) | No | Yes |
| PROGRAM | 'Y' | OMVS(PROGRAM (xx)) | Yes | Yes |
| | 'N' | OMVS(NOPROGRAM) | No | Yes |
| CPUTIMEMAX | 'Y' | OMVS(CPUTIMEMAX (xx)) | Yes | Yes |
| | 'N' | OMVS(NOCPUTIMEMAX) | No | Yes |
| ASSIZEMAX | 'Y' | OMVS(ASSIZEMAX (xx)) | Yes | Yes |
| | 'N' | OMVS(NOASSIZEMAX) | No | Yes |
| FILEPROCMAX | 'Y' | OMVS(FILEPROCMAX (xx)) | Yes | Yes |
| | 'N' | OMVS(NOFILEPROCMAX) | No | Yes |
| PROCUSERMAX | 'Y' | OMVS(PROCUSERMAX (xx)) | Yes | Yes |
| | 'N' | OMVS(NOPROCUSERMAX) | No | Yes |
| THREADSMAX | 'Y' | OMVS(THREADSMAX (xx)) | Yes | Yes |
| | 'N' | OMVS(NOTHREADSMAX) | No | Yes |

*Table 27. OMVS Segment Fields (continued)*

| Field Name | Flag Byte Values | ADDUSER/ALTUSER Keyword Reference | Allowed on Add Requests | Allowed on Alter Requests |
|---|---|---|---|---|
| MMAPAREAMAX | 'Y' | OMVS(MMAPAREAMAX (xx)) | Yes | Yes |
| | 'N' | OMVS(NOMMAPAREAMAX) | No | Yes |

*Table 28. TSO Segment Fields*

| Field Name | Flag Byte Values | ADDUSER/ALTUSER Keyword Reference | Allowed on Add Requests | Allowed on Alter Requests |
|---|---|---|---|---|
| ACCTNUM | 'Y' | TSO( ACCTNUM (xx)) | Yes | Yes |
| | 'N' | TSO (NOACCTNUM ) | No | Yes |
| DEST | 'Y' | TSO (DEST (xx)) | Yes | Yes |
| | 'N' | TSO( NODEST) | No | Yes |
| HLDCLASS | 'Y' | TSO( HOLDCLASS (xx)) | Yes | Yes |
| | 'N' | TSO( NOHOLDCLASS) | No | Yes |
| JOBCLASS | 'Y' | TSO( JOBCLASS (xx)) | Yes | Yes |
| | 'N' | TSO( NOJOBCLASS ) | No | Yes |
| MSGCLASS | 'Y' | TSO( MSGCLASS (xx)) | Yes | Yes |
| | 'N' | TSO( NOMSGCLASS ) | No | Yes |
| PROC | 'Y' | TSO( PROC(xx)) | Yes | Yes |
| | 'N' | TSO( NOPROC) | No | Yes |
| SIZE | 'Y' | TSO( SIZE (xx)) | Yes | Yes |
| | 'N' | TSO( NOSIZE) | No | Yes |
| MAXSIZE | 'Y' | TSO(MAXSIZE (xx)) | Yes | Yes |
| | 'N' | TSO(NOMAXSIZE) | No | Yes |
| SYSOUTCL | 'Y' | TSO( SYSOUTCL (xx)) | Yes | Yes |
| | 'N' | TSO(NOSYSOUTCL) | No | Yes |
| USERDATA | 'Y' | TSO( USERDATA (xx)) | Yes | Yes |
| | 'N' | TSO(NOUSERDATA) | No | Yes |
| UNIT | 'Y' | TSO( UNIT (xx)) | Yes | Yes |
| | 'N' | TSO(NOUNIT ) | No | Yes |
| COMMAND | 'Y' | TSO( COMMAND (xx)) | Yes | Yes |
| | 'N' | TSO(NOCOMMAND) | No | Yes |
| SECLABEL | 'Y' | TSO( SECLABEL (xx)) | Yes | Yes |
| | 'N' | TSO( NOSSECLABEL ) | No | Yes |

*Table 29. CICS Segment Fields*

| Field Name | Flag Byte Values | ADDUSER/ALTUSER Keyword Reference | Allowed on Add Requests | Allowed on Alter Requests |
|---|---|---|---|---|
| OPIDENT | 'Y' | CICS(OPIDENT (xx)) | Yes | Yes |
| | 'N' | CICS(NOOPIDENT) | No | Yes |

*Table 29. CICS Segment Fields (continued)*

| Field Name | Flag Byte Values | ADDUSER/ALTUSER Keyword Reference | Allowed on Add Requests | Allowed on Alter Requests |
|---|---|---|---|---|
| OPCLASS | 'Y' | CICS( OPCLASS (xx ...)) | Yes | Yes |
| | 'A' | CICS( ADDOPCLASS (xx ... )) | No | Yes |
| | 'D' | CICS(DELOPCLASS (xx ...)) | No | Yes |
| | 'N' | CICS(NOOPCLASS ) | No | Yes |
| OPPRTY | 'Y' | CICS(OOPPRTY (xx)) | Yes | Yes |
| | 'N' | CICS( NOOPPRTY) | No | Yes |
| TIMEOUT | 'Y' | CICS( TIMEOUT (xx)) | Yes | Yes |
| | 'N' | CICS( NOTIMEOUT) | No | Yes |
| XRFSOFF | 'Y' | CICS( XRFSOFF (xx)) | Yes | Yes |
| | 'N' | CICS( NOXRFSOFF) | No | Yes |

*Table 30. NetView Segment Fields*

| Field Name | Flag Byte Values | ADDUSER/ALTUSER Keyword Reference | Allowed on Add Requests | Allowed on Alter Requests |
|---|---|---|---|---|
| CONSNAME | 'Y' | NETVIEW( CONSNAME (xx)) | Yes | Yes |
| | 'N' | NETVIEW( NOCONSNAME) | No | Yes |
| CTL | 'Y' | NETVIEW( CTL (xx)) | Yes | Yes |
| | 'N' | NETVIEW( NOCTL) | No | Yes |
| DOMAINS | 'Y' | NETVIEW( DOMAINS (xx ...)) | Yes | Yes |
| | 'A' | NETVIEW( ADDDOMAINS (xx ...)) | No | Yes |
| | 'D' | NETVIEW( DELDOMAINS (xx ...)) | No | Yes |
| | 'N' | NETVIEW( NODOMAINS (xx ...)) | No | Yes |
| IC | 'Y' | NETVIEW(IC (xx )) | Yes | Yes |
| | 'N' | NETVIEW( NOIC) | No | Yes |
| MSGRECVR (Boolean) | 'Y' | NETVIEW( MSGRECVR (YES)) | Yes | Yes |
| | 'N' | NETVIEW(MSGRECVR (NO)) | Yes | Yes |
| NGMFADMN (Boolean) | 'Y' | NETVIEW( NGMFADMN(YES)) | Yes | Yes |
| | 'N' | NETVIEW( NGMFADMN (NO)) | No | Yes |

*Table 30. NetView Segment Fields (continued)*

| Field Name | Flag Byte Values | ADDUSER/ALTUSER Keyword Reference | Allowed on Add Requests | Allowed on Alter Requests |
|---|---|---|---|---|
| NGMFVSPN | 'Y' | NETVIEW (NGMFVSPN (xx)) | Yes | Yes |
| | 'N' | NETVIEW( NONGMFVSPN) | No | Yes |
| OPCLASS | 'Y' | NETVIEW(OPCLASS (xx ...)) | Yes | Yes |
| | 'A' | NETVIEW( ADDOPCLASS (xx ...)) | No | Yes |
| | 'D' | NETVIEW( DELOPCLASS (xx ...)) | No | Yes |
| | 'N' | NETVIEW( NOOPCLASS) | No | Yes |

*Table 31. DCE Segment Fields*

| Field Name | Flag Byte Values | ADDUSER/ALTUSER Keyword Reference | Allowed on Add Requests | Allowed on Alter Requests |
|---|---|---|---|---|
| AUTOLOG (Boolean) | 'Y' | DCE(AUTOLOGIN (YES)) | Yes | Yes |
| | 'N' | DCE(AUTOLOGIN(NO)) | Yes | Yes |
| DCENAME | 'Y' | DCE(DCENAME(xx)) | Yes | Yes |
| | 'N' | DCE(DCENAME ) | No | Yes |
| HOMECELL | 'Y' | DCE(HOMECELL (xx)) | Yes | Yes |
| | 'N' | DCE(NOHOMECELL) | No | Yes |
| HOMEUUID | 'Y' | DCE(HOMEUUID (xx)) | Yes | Yes |
| | 'N' | DCE(NOHOMEUUID) | No | Yes |
| UUID | 'Y' | DCE(UUID(xx)) | Yes | Yes |
| | 'N' | DCE(NOUUID) | No | Yes |

*Table 32. DFP Segment Fields*

| Field Name | Flag Byte Values | ADDUSER/ALTUSER Keyword Reference | Allowed on Add Requests | Allowed on Alter Requests |
|---|---|---|---|---|
| DATAAPPL | 'Y' | DFP(DATAAPPL(xx)) | Yes | Yes |
| | 'N' | DFP(NODATAAPPL) | No | Yes |
| DATACLAS | 'Y' | DFP(DATACLAS(xx)) | Yes | Yes |
| | 'N' | DFP(NODATACLAS) | No | Yes |
| MGMCLAS | 'Y' | DFP(MGMTCLAS(xx)) | Yes | Yes |
| | 'N' | DFP(NOMGMTCLAS) | No | Yes |
| STORCLAS | 'Y' | DFP(STORCLAS(XX)) | Yes | Yes |
| | 'N' | DFP(NOSTORCLAS) | No | Yes |

*Table 33. Language Segment Fields*

| Field Name | Flag Byte Values | ADDUSER/ALTUSER Keyword Reference | Allowed on Add Requests | Allowed on Alter Requests |
|---|---|---|---|---|
| Primary | 'Y' | LANGUAGE (PRIMARY(xx)) | Yes | Yes |
| | 'N' | LANGUAGE(NOPRIMARY) | No | Yes |
| Second | 'Y' | LANGUAGE(SECONDARY(xx)) | Yes | Yes |
| | 'N' | LANGUAGE(NOSECONDARY) | No | Yes |

*Table 34. OPERPARM Segment Fields*

| Field Name | Flag Byte Values | ADDUSER/ALTUSER Keyword Reference | Allowed on Add Requests | Allowed on Alter Requests |
|---|---|---|---|---|
| ALTGRP | 'Y' | OPERPARM (ALTGRP(xx)) | Yes | Yes |
| | 'N' | OPERPARM(NOALTGRP) | No | Yes |
| OPERAUTH | 'Y' | OPERPARM(AUTH(xx)) | Yes | Yes |
| | 'N' | OPERPARM(NOAUTH) | No | Yes |
| AUTO | 'Y' | OPERPARM(AUTO(xx)) | Yes | Yes |
| | 'N' | OPERPARM(NOAUTO)) | No | Yes |
| CMDSYS | 'Y' | OPERPARM(CMDSYS (xx)) | Yes | Yes |
| | 'N' | OPERPARM (NOCMDSYS) | No | Yes |
| DOM | 'Y' | OPERPARM(DOM xx)) | Yes | Yes |
| | 'N' | OPERPARM (NODOM) | No | Yes |
| KEY | 'Y' | OPERPARM(KEY(xx)) | Yes | Yes |
| | 'N' | OPERPARM(NOKEY) | No | Yes |
| LEVEL | 'Y' | OPERPARM(LEVEL (xx)) | Yes | Yes |
| | 'N' | OPERPARM(NOLEVEL) | No | Yes |
| LOGCMD | 'Y' | OPERPARM(LOGCMDRESP(xx)) | Yes | Yes |
| | 'N' | OPERPARM(NOLOGCMDRESP) | No | Yes |
| MFORM | 'Y' | OPERPARM(MFORM(xx)) | Yes | Yes |
| | 'N' | OPERPARM(NOMFORM) | No | Yes |
| MIGID | 'Y' | OPERPARM(MIGID(xx)) | Yes | Yes |
| | 'N' | OPERPARM(NOMIGID) | No | Yes |
| MONITOR | 'Y' | OPERPARM(xx)) | Yes | Yes |
| | 'N' | OPERPARM(NOMONITOR) | No | Yes |
| MSCOPE | 'Y' | OPERPARM(MSCOPE(xx ...)) | Yes | Yes |
| | 'A' | OPERPARM(ADDMSCOPE(xx ...)) | No | Yes |
| | 'D' | OPERPARM(DELMSCOPE(xx ...)) | No | Yes |
| | 'N' | OPERPARM(NOMSCOPE) | No | Yes |
| ROUTCODE | 'Y' | OPERPARM(ROUTCODE(xx ...) | Yes | Yes |
| | 'N' | OPERPARM(NOROUTCODE) | No | Yes |
| STORAGE | 'Y' | OPERPARM(STORAGE(xx)) | Yes | Yes |
| | 'N' | OPERPARM(NOSTORAGE) | No | Yes |

*Table 34. OPERPARM Segment Fields (continued)*

| Field Name | Flag Byte Values | ADDUSER/ALTUSER Keyword Reference | Allowed on Add Requests | Allowed on Alter Requests |
|---|---|---|---|---|
| UD | 'Y' | OPERPARM( UD (xx)) | Yes | Yes |
| | 'N' | OPERPARM(NOUD) | No | Yes |

*Table 35. OVM Segment Fields*

| Field Name | Flag Byte Values | ADDUSER/ALTUSER Keyword Reference | Allowed on Add Requests | Allowed on Alter Requests |
|---|---|---|---|---|
| FSROOT | 'Y' | OVM(FSROOT (xx)) | Yes | Yes |
| | 'N' | OVM(NOFSROOT) | No | Yes |
| VHOME | 'Y' | OVM(HOME(xx)) | Yes | Yes |
| | 'N' | OVM(NOHOME) | No | Yes |
| VPROGRAM | 'Y' | OVM(PROGRAM(xx)) | Yes | Yes |
| | 'N' | OVM(NOPROGRAM) | No | Yes |
| VUID | 'Y' | OVM(UID(xx)) | Yes | Yes |
| | 'N' | OVM(NOUID) | No | Yes |

*Table 36. WORKATTR Segment Fields*

| Field Name | Flag Byte Values | ADDUSER/ALTUSER Keyword Reference | Allowed on Add Requests | Allowed on Alter Requests |
|---|---|---|---|---|
| WAACCNT | 'Y' | WORKATTR(WAACCNT (xx)) | Yes | Yes |
| | 'N' | WORKATTR(NOWAACCNT) | No | Yes |
| WAADDR1 | 'Y' | WORKATTR(WAADDR1 (xx)) | Yes | Yes |
| | 'N' | WORKATTR(NOWADDR1) | No | Yes |
| WAADDR2 | 'Y' | WORKATTR(WAADDR2 (xx)) | Yes | Yes |
| | 'N' | WORKATTR(NOWADDR2) | No | Yes |
| WAADDR3 | 'Y' | WORKATTR(WAADDR3 (xx)) | Yes | Yes |
| | 'N' | WORKATTR(NOWADDR3) | No | Yes |
| WAADDR4 | 'Y' | WORKATTR(WAADDR4 (xx)) | Yes | Yes |
| | 'N' | WORKATTR(NOWADDR4) | No | Yes |
| WABLDG | 'Y' | WORKATTR(WABLDG(xx)) | Yes | Yes |
| | 'N' | WORKATTR(NOWABLDG) | No | Yes |
| WADEPT | 'Y' | WORKATTR(WADEPT (xx)) | Yes | Yes |
| | 'N' | WORKATTR(NOWADEPT) | No | Yes |
| WANAME | 'Y' | WORKATTR(WANAME (xx)) | Yes | Yes |
| | 'N' | WORKATTR(NOWANAME(xx)) | No | Yes |
| WAROOM | 'Y' | WORKATTR(WAROOM (xx)) | Yes | Yes |
| | 'N' | WORKATTR(NOWAROOM) | No | Yes |

*Table 37. LNOTES Segment Fields*

| Field Name | Flag Byte Values | ADDUSER/ALTUSER Keyword Reference | Allowed on Add Requests | Allowed on Alter Requests |
|---|---|---|---|---|
| SNAME | 'Y' | LNOTES(SNAME (xx)) | Yes | Yes |
| | 'N' | LNOTES(NOSNAME) | No | Yes |

*Table 38. NDS Segment Fields*

| Field Name | Flag Byte Values | ADDUSER/ALTUSER Keyword Reference | Allowed on Add Requests | Allowed on Alter Requests |
|---|---|---|---|---|
| UNAME | 'Y' | NDS(UNAME (xx)) | Yes | Yes |
| | 'N' | NDS(NOUNAME) | No | Yes |

*Table 39. KERB Segment Fields*

| Field Name | Flag Byte Values | ADDUSER/ALTUSER Keyword Reference | Allowed on Add Requests | Allowed on Alter Requests |
|---|---|---|---|---|
| KERBNAME | 'Y' | KERB(KERBNAME(xx)) | Yes | Yes |
| | 'N' | KERB(NOKERBNAME) | No | Yes |
| MAXTKTLF | 'Y' | KERB(MAXTKTLFE(xx)) | Yes | Yes |
| | 'N' | KERB(NOMAXTKTLFE) | No | Yes |

## Group Administration

For the ADMN_ADD_GROUP, ADMN_DEL_GROUP, ADMN_ALT_GROUP, and ADMN_LST_GROUP function codes, the mapping associated with the function specific parameter list is mapped as follows:

| Offset | Length | Description |
|---|---|---|
| 0 | 1 | Length of the Group Name |
| 1 | 8 | Upper case RACF Group Name |
| 9 | 1 | Reserved |
| 10 | 2 | Output offset to the segment or field entry in error in relation to the start of the Parm_list. Only applied to ADMN_ADD_GROUP and ADMN_ALT_GROUP requests when an "invalid 'request" error is returned to the caller. |
| 12 | 2 | Number of RACF profile segments |
| 14 | * | Start of first segment entry |

For ADMN_DEL_GROUP, no segment data is expected. The number of segments should be zero. If non-zero, any segment data present is ignored.

Each segment entry is mapped as follows:

| Offset | Length | Description |
|---|---|---|
| 0 | 8 | Profile segment name (left justified, uppercase, and padded with blanks). For ADMN_LST_GROUP, any GROUP profile segment as defined in the RACF database templates is acceptable. For ADMN_ADD_GROUP and ADMN_ALT_GROUP, the following segments are supported BASE: DFP, OMVS, OVM, and TME. |
| 8 | 1 | Flag byte. ″Y′ to create (or alter) the segment. ′N′ delete (or not create) the segment. |
| 9 | 2 | Number of fields to update within a segment |
| 11 | * | First field for segment |

For ADMN_LST_GROUP, the flag byte is ignored.

For ADMN_ADD_GROUP, the flag byte indicates whether the segment should be created (′Y′) or not (′N′). Because the BASE segment must always be created, the flag byte for BASE is ignored. In addition, the flag byte of ′I′ may be specified for any segment name. When the callable service finds the ′I′ flag byte, it ignores the segment.

For ADMN_ALT_GROUP, the flag byte indicates whether the segment should be altered (′Y′) or should no longer exist (′N′). Because the BASE segment cannot be deleted, the flag byte for BASE is ignored. In addition, the flag byte of ′I′ may be specified for any segment name. When the callable service finds the ′I′ flag byte, it ignores the segment.

For ADMN_LST_USER, ADMN_ADD_GROUP, ADMN_ALT_GROUP with flag byte ′N′, no field data is expected. The number of fields should be zero. If non-zero, any field data present is ignored.

Each field entry is mapped as follows:

| Offset | Length | Description |
|---|---|---|
| 0 | 8 | Field name as defined below. All field names must be left justified, entered in all uppercase, and padded with blanks. |
| 8 | 1 | Field specific flag byte |
| 9 | 2 | Length of field data |
| 11 | * | Field data |

For ADMN_ADD_GROUP and ADMN_ALT_GROUP, the following rules apply:

- For boolean fields, the flag byte value of 'Y' or 'N' determines the flag setting. No field data is allowed for boolean fields. Coding field data results in an "Invalid Request" error being returned to the caller.
- For text fields, the flag byte 'Y' indicates the field should be created (or altered) with the specified field data. The flag byte 'N' indicates the field should be deleted (or not created). For flag byte 'N', any field data specified is ignored.
- For repeating text fields, a flag byte of 'A' indicates that the specified field data should be added to the existing data (if any). Field data must be specified. If field data is not specified, an "Invalid Request" error is returned to the caller. The flag byte 'D' indicates that the specified data should be deleted (if any).

In addition to the flag bytes specified in the following tables, a flag byte of 'I' may be specified for any segment name. The 'I' flag byte indicates to the callable service to ignore the field. Any field data specified with a flag byte of 'I' is ignored. In addition, the flag byte of 'I' may be specified for any segment name. When the callable service finds the 'I' flag byte, it ignores the segment.

The following tables define the field keywords and their usage. All field names relate directly to ADDGROUP/ALTGROUP keywords. If you have any questions about field usage and data, refer to *OS/390 SecureWay Security Server RACF Command Language Reference*.

*Table 40. BASE Segment Fields*

| Field Name | Flag Byte Values | ADDGROUP/ALTGROUP Keyword Reference | Allowed on Add Requests | Allowed on Alter Requests |
|---|---|---|---|---|
| SUPGROUP | 'Y' | SUPGROUP(xx) | Yes | Yes |
| OWNER | 'Y' | OWNER(xx) | Yes | Yes |
| TERMUACC (Boolean) | 'Y' | TERMUACC | Yes | Yes |
| | 'N' | NOTERMUACC | Yes | Yes |
| DATA | 'Y' | DATA(xx) | Yes | Yes |
| | 'N' | NODATA | No | Yes |
| MODEL | 'Y' | MODEL (xx) | Yes | Yes |
| | 'N' | NOMODEL | No | Yes |

*Table 41. DFP Segment Fields*

| Field Name | Flag Byte Values | ADDGROUP/ALTGROUP Keyword Reference | Allowed on Add Requests | Allowed on Alter Requests |
|---|---|---|---|---|
| DATAAPPL | 'Y' | DFP(DATAAPPL(xx)) | Yes | Yes |
| | 'N' | DFP(NODATAAPPL) | No | Yes |
| DATACLAS | 'Y' | DFP(DATACLAS(xx)) | Yes | Yes |
| | 'N' | DFP(NODATACLAS) | No | Yes |
| MGMTCLAS | 'Y' | DFP(MGMTCLAS(xx)) | Yes | Yes |
| | 'N' | DFP(NOMGMTCLAS) | No | Yes |
| STORCLAS | 'Y' | DFP(STORCLAS(xx)) | Yes | Yes |
| | 'N' | DFP(NOSTORCLAS) | No | Yes |

*Table 42. OMVS Segment Fields*

| Field Name | Flag Byte Values | ADDGROUP/ALTGROUP Keyword Reference | Allowed on Add Requests | Allowed on Alt Requests |
|---|---|---|---|---|
| GID | 'Y' | OMVS(GID(xx)) | Yes | Yes |
| | 'N' | OMVS(NOGID) | No | Yes |

*Table 43. OVM Segment Fields*

| Field Name | Flag Byte Values | ADDGROUP/ALTGROUP Keyword Reference | Allowed on Add Requests | Allowed on Alt Requests |
|---|---|---|---|---|
| GID | 'Y' | OVM(GID(xx)) | Yes | Yes |
| | 'N' | OVM(NOGID) | No | Yes |

*Table 44. TME Segment Fields*

| Field Name | Flag Byte Values | ADDGROUP/ALTGROUP Keyword Reference | Allowed on Add Requests | Allowed on Alter Requests |
|---|---|---|---|---|
| ROLES | 'Y' | TME(ROLES(xx ...)) | Yes | Yes |
| | 'A' | TME(ADDROLES(xx ...)) | No | Yes |
| | 'D' | TME(DELROLES (xx ...)) | No | Yes |
| | 'N' | TME(NOROLES) | No | Yes |

**Group Connection Administration:** For the ADMN_CONNECT and ADMN_REMOVE function codes the mapping associated with the function specific parameter list is mapped as follows:

| Offset | Length | Description |
|---|---|---|
| 0 | 1 | Length of the User ID1 |
| 1 | 8 | Upper case RACF User ID |
| 9 | 1 | Reserved |
| 10 | 2 | Output offset to the field entry in error in relationship to the start of the Parm_list. Only applies to ADMN_CONNECT and ADMN_REMOVE requests when an "Invalid Request" error is returned to the caller. |
| 12 | 2 | Number of RACF segments |
| 14 | * | Start of first segment entry |

Each segment entry is mapped as follows:

| Offset | Length | Description |
|---|---|---|
| 0 | 1 | Profile segment name (only the BASE segment is accepted for ADMN_CONNECT and ADMN_REMOVE) |
| 1 | 8 | Flag byte. 'Y' to create (or alter) the segment. 'N' delete (or not create) the segment |
| 9 | 2 | Number of fields to update within a segment. |

| Offset | Length | Description |
|---|---|---|
| 11 | * | First field for segment |

By convention, use the BASE segment to specify field information for ADMN_CONNECT and ADMN_REMOVE. The flag byte is ignored. Each field entry is mapped as follows:

| Offset | Length | Description |
|---|---|---|
| 0 | 1 | Field name as defined below. All field names must be left justified, entered in all uppercase, and padded with blanks. |
| 1 | 8 | Field specific flag byte.. |
| 9 | 2 | Length of field data |
| 11 | * | Field data |

For ADMN_CONNECT and ADMN_REMOVE, the following rules apply:

- For boolean fields, the flag byte value of 'Y' or 'N' determines the flag setting. No field data is allowed for boolean fields. Coding field data results in an "Invalid Request" error being returned to the caller.
- For text fields, the flag byte 'Y' indicates the field should be created (or altered) with the specified field data. The flag byte 'N' indicates the field should be deleted (or not created). For flag byte 'N', any field data specified is ignored.
- For repeating text fields, a flag byte of 'A' indicates that the specified field data should be added to the existing data (if any). Field data must be specified. If field data is not specified, an "Invalid Request" error is returned to the caller. The flag byte 'D' indicates that the specified data should be deleted (if any).

In addition to the flag bytes specified in the following tables, a flag byte of 'I' may be specified for any segment name. The 'I' flag byte indicates to the callable service to ignore the field. Any field data specified with a flag byte of 'I' is ignored.

The following table defines the field keywords and their usage. All field names relate directly to the CONNECT and REMOVE keywords. If you have any questions about the field usage and data of these commands, see *OS/390 SecureWay Security Server RACF Command Language Reference*.

*Table 45. Base Segment Fields*

| Field Name | Flag Byte Values | CONNECT and REMOVE Keyword Reference | Allowed on CONNECT Requests | Allowed on REMOVE requests |
|---|---|---|---|---|
| GROUP | 'Y' | GROUP(xx) | Yes | Yes |
| OWNER | 'Y' | OWNER (xx) | Yes | Yes |
| ADSP (Boolean) | 'Y' | ADSP | Yes | No |
| | 'N' | NOADSP | Yes | No |
| AUDITOR (Boolean) | 'Y' | AUDITOR | Yes | No |
| | 'N' | NOAUDITOR | Yes | No |
| GRPACC (Boolean) | 'Y' | GRPACC | Yes | No |
| | 'N' | NOGRPACC | Yes | No |

*Table 45. Base Segment Fields (continued)*

| Field Name | Flag Byte Values | CONNECT and REMOVE Keyword Reference | Allowed on CONNECT Requests | Allowed on REMOVE requests |
|---|---|---|---|---|
| OPER (Boolean) | 'Y' | OPERATIONS | Yes | No |
| | 'N' | NOOPERATIONS | Yes | No |
| REVOKE | 'Y' | REVOKE (xx) | Yes | No |
| RESUME | 'Y' | RESUME (xx) | Yes | No |
| SPECIAL (Boolean) | 'Y' | SPECIAL | Yes | No |
| | 'N' | NOSPECIAL | Yes | No |
| UACC | 'Y' | UACC (xx) | Yes | No |

**Resource Profile Administration:** For all of the resource-related function codes (ADMN_ADD_GENRES, ADMN_ALT_GENRES, ADMN_DEL_GENRES, ADMN_LST_GENRES, ADMN_ADD_DS, ADMN_ALT_DS, ADMN_DEL_DS, ADMN_LST_DS, and ADMN_PERMIT), the mapping associated with the function-specific parameter list is mapped as follows:

| Offset | Length | Description |
|---|---|---|
| 0 | 1 | Length of the class name |
| 1 | 8 | Upper case RACF class name |
| NOTE: The class name is not required for the data set functions. | | |
| 9 | 1 | Reserved |
| 10 | 2 | Output offset to the segment or field entry in error in relation to the start of the Parm_list. Only applies to add, alter, and list requests when an "Invalid Request" error is returned to the caller. |
| 12 | 2 | Number of RACF profile segments |
| 14 | * | Start of the first segment entry |

Each segment entry is mapped as follows:

| Offset | Length | Description |
|---|---|---|
| 0 | 8 | Profile segment name (left justified, uppercase, and padded with blanks). By convention, use the BASE segment to specify field information for ADMN_PERMIT and the delete and list functions (specify the NORACF field if you do not want the BASE segment listed). To list additional segment information, specify additional segment entries, where any general resource or data set profile segment as defined in the RACF database templates is acceptable. For ADMN_ADD_GENRES and ADMN_ALT_GENRES, the following segments are supported: BASE, DLFDATA, KERB, SESSION, SSIGNON, STDATA, SVFMR, and TME. For ADMN_ADD_DS and ADMN_ALT_DS, the following segments are supported: BASE, DFP, and TME. |

| Offset | Length | Description |
|---|---|---|
| 8 | 1 | Flag byte. 'Y' to create (or alter) the segment. 'N' delete (or not create) the segment. |
| 9 | 2 | Number of fields to update within a segment |
| 11 | * | First field for segment |

For ADMN_PERMIT, ADMN_DEL_DS, and for the list functions, the flag byte is ignored.

For the add functions, the flag byte indicates whether the segment should be create ('Y') or not ('N'). Because the BASE segment must always be created, the flag byte for BASE is ignored. In addition, the flag byte of 'I' may be specified for any segment name. When the callable service finds the 'I' flag byte, it ignores the segment.

For the alter functions, the flag byte indicates whether the segment should be altered ('Y') or should no longer exist ('N'). Because the base segment cannot be deleted, the flag byte for BASE is ignored. In addition, the flag byte of 'I' may be specified for any segment name. When the callable service finds the 'I' flag byte, it ignores the segment.

For the add and alter functions with flag byte 'N', no field data is expected. The number of fields should be zero. If non-zero, any field data present is ignored.

Each field entry is mapped as follows:

| Offset | Length | Description |
|---|---|---|
| 0 | 8 | Field name as defined below. All field names must be left justified, entered in all uppercase, and padded with blanks |
| 8 | 1 | Field-specific flag byte |
| 9 | 2 | Length of field data |
| 11 | * | Field data |

For boolean fields, the flag byte value of 'Y' or 'N' determines the flag setting. No field data is allowed for boolean fields. Coding field data results in an "Invalid Request" error being returned to the caller.

For the add and alter functions, the following rules apply:

- For text fields, the flag byte 'Y' indicates the field should be created (or altered) with the specified field data. The flag byte 'N' indicates the field should be deleted (or not created). For flag byte 'N", any field data specified is ignored.
- For repeating text fields, a flag byte of 'A' indicates that the specified field data is be added to the existing data (if any). Field data must be specified. If field data is not specified an "Invalid Request" error is returned to the caller. The flag byte 'D' indicates that the specified data (if any) is to be deleted.

In addition to the flag bytes specified in the following tables, the flag byte 'I' can be specified for any segment name. The 'I' flag byte indicates to the callable service to ignore the field. Any field data specified with a flag byte of 'I' is ignored.

The general resource, data set, and permit functions each utilize a separate set of field definitions. Table 46 shows where to find the field definitions for the resource related function codes.

*Table 46. Resource Related Field Definitions*

| For Field Definitions | See |
|---|---|
| ADMN_ADD_GESRES<br>ADMN_ALT_GENRES<br>ADMN_LST_GENRES | "General Resource Field Definitions". |
| NOTE: For ADMN_DEL_GENRES, specify only the PROFILE field in the BASE segment. | |
| ADMN_ADD_DS, ADMN_ALT_DS,<br>ADMN_LST_DS, ADMN_DEL_DS | "Data Set Field Definitions" on page 88 |
| ADMN_PERMIT | "Permit Field Definitions" on page 91 |

*General Resource Field Definitions:* The following tables define the general resource field keywords and their usage. All field names relate directly to the RDEFINE, RALTER, and RLIST keywords. If you have questions about the field usage and data, see *OS/390 SecureWay Security Server RACF Command Language Reference*.

*Table 47. BASE Segment Fields*

| Field Name | Flag Byte Value | RDEFINE/RALTER/RLIST Keyword Reference | Allowed on Add Requests | Allowed on Alter Requests | Allowed on List Requests |
|---|---|---|---|---|---|
| PROFILE | 'Y' | N/A (see note) | Yes | Yes | Yes |
| NOTE: This field is required. There is no associated command keyword since it is a positional parameter. For ADMN_DEL_GENRES, this is the only allowable field. | | | | | |
| CATEGORY | 'Y' | ADDCATEGORY(xx ...) | Yes | No | No |
| | 'A' | ADDCATEGORY(xx ...) | No | Yes | No |
| | 'D' | DELCATEGORY(xx ...) | No | Yes | No |
| NOTE: To remove unknown categories from the profile, specify the 'D' flag and a field length of zero. | | | | | |
| MEMBER | 'Y' | ADDMEM(xx ...) | Yes | No | No |
| | 'N' | ADDMEM(xx ...) | No | Yes | No |
| | 'D' | DELMEM(xx ...) | No | Yes | No |
| VOLUME | 'A' | ADDVOL(xx ...) | No | Yes | No |
| | 'D' | DELVOL(xx ...) | No | Yes | No |
| APPLDATA | 'Y' | APPLDATA(xx) | Yes | Yes | No |
| | 'N' | NOAPPLDATA | No | Yes | No |
| DATA | 'Y' | DATA(xx) | Yes | Yes | No |
| | 'N' | NODATA | No | Yes | No |
| AUDALTR | 'Y' | AUDIT(xx (ALTER)) | Yes | Yes | No |
| AUDCNTL | 'Y' | AUDIT(xx (CONTROL)) | Yes | Yes | No |
| AUDREAD | 'Y' | AUDIT(xx (READ)) | Yes | Yes | No |
| AUDUPDT | 'Y' | AUDIT(xx (UPDATE)) | Yes | Yes | No |
| AUDNONE | 'Y' | AUDIT(NONE) | Yes | Yes | No |
| GAUDALTR | 'Y' | GLOBALAUDIT(xx (ALTER)) | No | Yes | No |

*Table 47. BASE Segment Fields (continued)*

| Field Name | Flag Byte Value | RDEFINE/RALTER/RLIST Keyword Reference | Allowed on Add Requests | Allowed on Alter Requests | Allowed on List Requests |
|---|---|---|---|---|---|
| GAUDCNTR | 'Y' | GLOBALAUDIT(xx (CONTROL)) | No | Yes | No |
| GAUDREAD | 'Y' | GLOBALAUDIT(xx (READ)) | No | Yes | No |
| GAUDUPDT | 'Y' | GLOBALAUDIT(xx UPDATE)) | No | Yes | No |
| GAUDNONE | 'Y' | GLOBALAUDIT (NONE) | No | Yes | No |
| FPROFILE | 'Y' | FROM(xx) | Yes | No | No |
| FCLASS | 'Y' | FCLASS(xx) | Yes | No | No |
| FVOLUME | 'Y' | FVOLUME(xx) | Yes | No | No |
| FGENERIC (Boolean) | 'Y' | FGENERIC | Yes | No | No |
| LEVEL | 'Y' | LEVEL(xx) | Yes | Yes | No |
| NOTIFY | 'Y' | NOTIFY(xx) | Yes | Yes | No |
| | 'N' | NONOTIFY | No | Yes | No |
| OWNER | 'Y' | OWNER(xx) | Yes | Yes | No |
| SECLABLE | 'Y' | SECLABEL (xx) | Yes | Yes | No |
| | 'N' | NOSECLABEL | No | Yes | No |
| SECLEVEL | 'Y' | SECLEVEL(xx) | Yes | Yes | No |
| | 'N' | NOSECLEVEL | No | Yes | No |
| SINGLDSN (Boolean) | 'Y' | SINGLEDSN | Yes | Yes | No |
| | 'N' | NOSINGLEDSN | No | Yes | No |
| TIMEZONE | 'Y' | TIMEZONE(xx) | Yes | Yes | No |
| | 'N' | NOTIMEZONE | No | Yes | No |
| TVTOC (Boolean) | 'Y' | TVTOC | Yes | Yes | Yes |
| | 'N' | NOTVTOC | No | Yes | No |
| UACC | 'Y' | UACC(xx) | Yes | Yes | No |
| WARNING (Boolean) | 'Y' | WARNING | Yes | Yes | No |
| | 'N' | NOWARNING | No | Yes | No |
| WHENDAYS | 'Y' | WHEN(DAYS(xx)) | Yes | Yes | No |
| WHENTIME | 'Y' | WHEN( TIME (xx ...)) | Yes | Yes | No |
| ALL (Boolean) | 'Y' | ALL | No | No | Yes |
| AUTHUSER (Boolean) | 'Y' | AUTHUSER | No | No | Yes |
| GENERIC | 'Y' | GENERIC | No | No | Yes |
| | 'N' | NOGENERIC | No | No | Yes |
| HISTORY (Boolean) | 'Y' | HISTORY | No | No | Yes |
| NORACF (Boolean) | 'Y' | NORACF | No | No | Yes |

*Table 47. BASE Segment Fields (continued)*

| Field Name | Flag Byte Value | RDEFINE/RALTER/RLIST Keyword Reference | Allowed on Add Requests | Allowed on Alter Requests | Allowed on List Requests |
|---|---|---|---|---|---|
| NOYOURAC (Boolean) | 'Y' | NOYOURACC | No | No | Yes |
| RESGROUP | 'Y' | RESGROUP | No | No | Yes |
| STATS (Boolean) | 'Y' | STATISTICS | No | No | Yes |

*Table 48. DLFDATA Segment Fields*

| Field Name | Flag Byte Values | RDEFINE/RALTER Keyword Reference | Allowed on Add Requests | Allowed on Alter Requests |
|---|---|---|---|---|
| RETAIN (Boolean) | 'Y' | DLFDATA(RETAIN(YES)) | Yes | Yes |
| | 'N' | DLFDATA (RETAIN(NO)) | Yes | Yes |
| JOBNAME | 'Y' | DLFDATA JOBNAMES (...)) | Yes | Yes |
| | 'A' | DLFDATA(ADDJOBNAMES(...)) | No | Yes |
| | 'D' | DLFDATA(DELJOBNAMES(...)) | No | Yes |
| | 'N' | DLFDATA(NOJOBNAMES) | No | Yes |

*Table 49. SESSION Segment Fields*

| Field Name | Flag Byte Value | RDEFINE/RALTER Keyword Reference | Allowed on Add Requests | Allowed on Alter Requests |
|---|---|---|---|---|
| CONVSEC | 'Y' | SESSION (CONVSEC(...)) | Yes | Yes |
| | 'N' | SESSION (NOCONVSEC)) | No | Yes |
| INTERVAL | 'Y' | SESSION(INTERVAL(...)) | Yes | Yes |
| | 'N' | SESSION (NOINTERVAL)) | No | Yes |
| LOCK | 'Y' | SESSION (LOCK) | Yes | Yes |
| | 'N' | SESSION (NOLOCK) | No | Yes |
| SESSKEY | 'Y' | SESSION ( SESSKEY(...)) | Yes | Yes |
| | 'N' | SESSION(NOSESSKEY)) | No | Yes |

*Table 50. SSIGNON Segment Fields*

| Field Name | Flag Byte Values | REDEFINE/RALTER Keyword Reference | Allowed on Add Requests | Allowed on Alter Requests |
|---|---|---|---|---|
| KEYMASK | 'Y' | SSIGNON (KEYMASKED(...)) | Yes | Yes |
| | 'N' | SSIGNON (NOKEYMASKED) | No | Yes |
| KEYCRYPT | 'Y' | SSIGNON (KEYENCRYPTED (...)) | Yes | Yes |
| | 'N' | SSIGNON(NOKEYENCRYPTED) | No | Yes |

*Table 51. STDATA Segment Fields*

| Field Name | Flag Byte Values | RDEFINE/RALTER Keyword Reference | Allowed on Add Requests | Allowed on Alter Requests |
|---|---|---|---|---|
| USER | 'Y' | STDATA(USER(...)) | Yes | Yes |
| | 'N' | STDATA(NOUSER)) | No | Yes |
| GROUP | 'Y' | STDATA(GROUP(...)) | Yes | Yes |
| | 'N' | STDATA(NOGROUP) | No | Yes |
| PRIVLEGE (Boolean) | 'Y' | STDATA(PRIVILEGED(YES)) | Yes | Yes |
| | 'N' | STDATA(PRIVILEGED(NO)) | Yes | Yes |
| TRACE (Boolean) | 'Y' | STDATA(TRACE(YES)) | Yes | Yes |
| | 'N' | STDATA(TRACE(NO)) | Yes | Yes |
| TRUSTED (Boolean) | 'Y' | STDATA(TRUSTED(YES)) | Yes | Yes |
| | 'N' | STDATA(TRUSTED(NO)) | Yes | Yes |

*Table 52. SVFMR Segment Fields*

| Field Name | Flag Byte Values | RDEFINE/RALTER Keyword Reference | Allowed on Add Requests | Allowed on Alter Requests |
|---|---|---|---|---|
| SCRIPT | 'Y' | SVFMR(SCRIPTNAME(...)) | Yes | Yes |
| | 'N' | SVFMR(NOSCRIPTNAME) | No | Yes |
| PARMNAME | 'Y' | SVFMR(PARMNAME(...)) | Yes | Yes |
| | 'N' | SVFMR(NOPARMNAME) | No | Yes |

*Table 53. TME Segment Fields*

| Field Name | Flag Byte Values | RDEFINE/RALTER Keyword Reference | Allowed on Add Requests | Allowed on Alter Requests |
|---|---|---|---|---|
| ROLES | 'Y' | TME(ROLES(xx ...)) | Yes | Yes |
| | 'A' | TME(ADDROLES(xx ...)) | No | Yes |
| | 'D' | TME(DELROLES(xx ...)) | No | Yes |
| | 'N' | TME(NOROLES) | No | Yes |
| GROUPS | 'Y' | TME(GROUPS(xx ...)) | Yes | Yes |
| | 'A' | TME(ADDGROUPS(xx ...)) | No | Yes |
| | 'D' | TME(DELGROUPS(xx ...)) | No | Yes |
| | 'N' | TME(NOGROUPS) | No | Yes |
| RESOURCE | 'Y' | TME(RESOURCE(xx ...)) | Yes | Yes |
| | 'A' | TME(ADDRESOURCE(xx ...)) | No | Yes |
| | 'D' | TME(DELRESOURCE(xx ...)) | No | Yes |
| | 'N' | TME(NORESOURCE) | No | Yes |
| CHILDREN | 'Y' | TME(CHILDREN(xx ...)) | Yes | Yes |
| | 'A' | TME(ADDCHILDREN(xx ...)) | No | Yes |
| | 'D' | TME(DELCHILDREN(xx ...)) | No | Yes |
| | 'N' | TME(NOCHILDREN) | No | Yes |

*Table 53. TME Segment Fields (continued)*

| Field Name | Flag Byte Values | RDEFINE/RALTER Keyword Reference | Allowed on Add Requests | Allowed on Alter Requests |
|---|---|---|---|---|
| PARENT | 'Y' | TME(PARENT(xx ...)) | Yes | Yes |
| | 'N' | TME(NOPARENT) | No | Yes |

*Table 54. KERB Segment Fields*

| Field Name | Flag Byte Values | RDEFINE/RALTER Keyword Reference | Allowed on Add Requests | Allowed on Alter Requests |
|---|---|---|---|---|
| KERBNAME | 'Y' | KERB(KERBNAME(xx)) | Yes | Yes |
| | 'N' | KERB(NOKERBNAME) | No | Yes |
| DEFTKTLF | 'Y' | KERB(DEFTKTLFE(xx)) | Yes | Yes |
| | 'N' | KERB(NODEFTKTLFE) | No | Yes |
| MAXTKTLF | 'Y' | KERB(MAXTKTLFE(xx)) | Yes | Yes |
| | 'N' | KERB(NOMAXTKTLFE) | No | Yes |
| MINTKTLF | 'Y' | KERB(MINTKTLFE(xx)) | Yes | Yes |
| | 'N' | KERB(NOMINTKTLFE) | No | Yes |
| PASSWORD | 'Y' | KERB(PASSWORD(xx)) | Yes | Yes |
| | 'N' | KERB(NOPASSWORD) | No | Yes |

*Data Set Field Definitions:* The following table defines the DATASET field keywords and their usage. All field names relate directly to ADDSD, ALTDSD, DELDSD, and LISTDSD keywords. If you have any questions about field usage and data, see *OS/390 SecureWay Security Server RACF Command Language Reference*.

*Table 55. BASE Segment Fields*

| Field Name | Flag Byte Value | ADDSD, ALTDSD, DELDSD, LISTDSD Keyword Reference | Allowed on Add Requests | Allowed on Alter Requests | Allowed on List Requests | Allowed on Delete Requests |
|---|---|---|---|---|---|---|
| PROFILE | 'Y' | for list, DATASET (xx...) | Yes | Yes | Yes | Yes |
| NOTE: For add, alter, and delete, this field is required. This is no associated command keyword because it is a positional parameter. For list, this field is optional and is used in the DATASET(xx...) keyword.<br><br>For add and alter, if working with a password-protected data set, append "/password? to the data set profile name, and include this additional data in the length field for the data set profile name. | | | | | | |
| CATEGORY | 'Y' | ADDCATEGORY (xx ...) | Yes | No | No | No |
| | 'A' | ADDCATEGORY (xx ...) | No | Yes | No | No |
| | 'D' | DELCATEGORY (xx ...) | No | Yes | No | No |

*Table 55. BASE Segment Fields (continued)*

| Field Name | Flag Byte Value | ADDSD, ALTDSD, DELDSD, LISTDSD Keyword Reference | Allowed on Add Requests | Allowed on Alter Requests | Allowed on List Requests | Allowed on Delete Requests |
|---|---|---|---|---|---|---|
| NOTE: To remove unknown categories from the profile, specify the 'D' flag and a field length of zero. | | | | | | |
| VOLUME | 'Y' | VOLUME (xx ...) | Yes | Yes | Yes | Yes |
| | 'A' | ADDVOL (xx ...) | No | Yes | No | |
| | 'D' | DELVOL (xx ...) | No | Yes | No | |
| ALTVOL | 'Y' | ALTVOL (xx) | No | Yes | No | No |
| GENERIC (Boolean) | 'Y' | GENERIC | Yes | Yes | Yes | Yes |
| | 'N' | NOGENERIC | No | No | Yes | No |
| MODEL (Boolean) | 'Y' | MODEL | Yes | No | No | No |
| TAPE (Boolean) | 'Y' | TAPE | Yes | No | No | No |
| SET (Boolean) | 'Y' | SET | Yes | Yes | No | Yes |
| SETONLY (Boolean) | 'Y' | SETONLY | Yes | No | No | No |
| AUDALTR | 'Y' | AUDIT(xx(ALTER)) | Yes | Yes | No | No |
| AUDCNTL | 'Y' | AUDIT (xx (CONTROL)) | Yes | Yes | No | No |
| AUDREAD | 'Y' | AUDIT (xx (READ)) | Yes | Yes | No | No |
| AUDUPDT | 'Y' | AUDIT ( xx (UPDATE)) | Yes | Yes | No | No |
| AUDNONE (Boolean) | 'Y' | AUDIT (NONE) | Yes | Yes | No | No |
| GAUDALTR | 'Y' | GLOBALAUDIT (xx (ALTER)) | No | Yes | No | No |
| GAUDCNTR | 'Y' | GLOBALAUDIT (xx (CONTROL)) | No | Yes | No | No |
| GAUDREAD | 'Y' | GLOBALAUDIT (xx (READ)) | No | Yes | No | No |
| GAUDUPDT | 'Y' | GLOBAL AUDIT (xx UPDATE)) | No | Yes | No | No |
| GAUDNONE (Boolean) | 'Y' | GLOBALAUDIT (NONE) | No | Yes | No | No |
| DATA | 'Y' | DATA (xx) | Yes | Yes | No | No |
| | 'N' | NODATA | No | Yes | No | No |
| ERASE (Boolean) | 'Y' | ERASE | Yes | Yes | No | No |
| | 'N' | NOERASE | No | Yes | No | No |
| FILESEQ | 'Y' | FILESEQ (xx) | Yes | No | No | No |
| FROM | 'Y' | FROM (xx) | Yes | No | No | No |

*Table 55. BASE Segment Fields (continued)*

| Field Name | Flag Byte Value | ADDSD, ALTDSD, DELDSD, LISTDSD Keyword Reference | Allowed on Add Requests | Allowed on Alter Requests | Allowed on List Requests | Allowed on Delete Requests |
|---|---|---|---|---|---|---|
| FCLASS | 'Y' | FCLASS (xx) | Yes | No | No | No |
| FVOLUME | 'Y' | FVOLUME (xx) | Yes | No | No | No |
| FGENERIC (Boolean) | 'Y' | FGENERIC | Yes | No | No | No |
| LEVEL | 'Y' | LEVEL (xx) | Yes | Yes | No | No |
| NOTIFY | 'Y' | NOTIFY (xx) | Yes | Yes | No | No |
| | 'N' | NONOTIFY | No | Yes | No | No |
| OWNER | 'Y' | OWNER (xx) | Yes | Yes | No | No |
| SECLABLE | 'Y' | SECLABEL (xx) | Yes | Yes | No | No |
| | 'N' | NOSECLABEL | No | Yes | No | No |
| SECLEVEL | 'Y' | SECLEVEL (xx) | Yes | Yes | No | No |
| | 'N' | NOSECLEVEL | No | Yes | No | No |
| UACC | 'Y' | UACC | Yes | Yes | No | No |
| UNIT | 'Y' | UNIT (xx) | Yes | Yes | No | No |
| WARNING (Boolean) | 'Y' | WARNING | Yes | Yes | No | No |
| | 'N' | NOWARNING | No | Yes | No | No |
| PREFIX | 'Y' | PREFIX (xx) | No | No | Yes | No |
| ALL (Boolean) | 'Y' | ALL | No | No | Yes | No |
| AUTHUSER (Boolean) | 'Y' | AUTHUSER | No | No | Yes | No |
| DSNS (Boolean) | 'Y' | DSNS | No | No | Yes | No |
| HISTORY | 'Y' | HISTORY | No | No | Yes | No |
| NORACF (Boolean) | 'Y' | NORACF | No | No | Yes | No |
| STATS (Boolean) | 'Y' | STATISTICS | No | No | Yes | No |
| RETPD | 'Y' | RETPD (xx) | Yes | Yes | No | No |

*Table 56. DFP Segment Fields*

| Field Name | Flag Byte Value | ADDSD and ALTDSD Keyword Reference | Allowed on Add Requests | Allowed on Alter Requests |
|---|---|---|---|---|
| RESOWNER | 'Y' | DFP(RESOWNER(xx)) | Yes | Yes |
| | 'N' | DFP(NORESOWNER) | No | Yes |

*Table 57. TME Segment Fields*

| Field Name | Flag Byte Value | ADDSD and ALTDSD Keyword Reference | Allowed on Add Requests | Allowed on Alter Requests |
|---|---|---|---|---|
| ROLES | 'Y' | TME(ROLES(xx...)) | Yes | Yes |
| | 'A' | TME(ADDROLES(xx...)) | No | Yes |
| | 'D' | TME(DELROLES(xx...)) | No | Yes |
| | 'N' | TME(NOROLES) | No | Yes |

*Permit Field Definitions:* The following table defines the permit field keywords and their usage. All field names relate directly to PERMIT keywords. For information on field usage and data, see *OS/390 SecureWay Security Server RACF Command Language Reference*.

*Table 58. Base Segment Fields*

| Field Name | Flag Byte Values | PERMIT Keyword Reference |
|---|---|---|
| PROFILE | 'Y' | N/A (See note.) |
| NOTE: This field is required; there is no associated command keyword since it is a positional parameter. | | |
| ACCESS | 'Y' | ACCESS (xx) |
| DELETE (Boolean) | 'Y' | DELETE |
| FPROFILE | 'Y' | FROM (xx) |
| FCLASS | 'Y' | FCLASS (xx) |
| FVOLUME | 'Y' | FVOLUME (xx) |
| FGENERIC (Boolean) | 'Y' | FGENERIC |
| GENERIC (Boolean) | 'Y' | GENERIC |
| ID | 'Y' | ID (xx) |
| RESET | 'Y' | RESET (xx) |
| VOLUME | 'Y' | VOLUME (xx) |
| WHENAPPC | 'Y' | WHEN (APPCPORT (...)) |
| WHENCONS | 'Y' | WHEN (CONSOLE ( ...)) |
| WHENJES | 'Y' | WHEN (JESINPUT (...)) |
| WHENPROG | 'Y' | WHEN (PROGRAM (...)) |
| WHENTERM | 'Y' | WHEN (TERMINAL (...)) |
| WHENSYS | 'Y' | WHEN (SYSID (...)) |

*Parameter List Format for SETROPTS Administration:* For the ADMN_ALT_SETR function code, the function-specific parameter list is mapped as follows:

*Table 59. Parameter List Mapping for SETROPTS Administration*

| Offset | Length | Description |
|---|---|---|
| 0 | 10 | Reserved |

*Table 59. Parameter List Mapping for SETROPTS Administration (continued)*

| Offset | Length | Description |
|---|---|---|
| 10 | 2 | Output offset to the segment or field entry in error, relative to the start of the Parm_list. Only applies when an "Invalid Request" error is returned to the caller. |
| 12 | 2 | Number of RACF profile segments |
| 14 | * | Start of first segment entry |

The segment entry is mapped as follows:

*Table 60. Segment Entry Fields*

| Offset | Length | Description |
|---|---|---|
| 0 | 8 | Profile segment name (only the "BASE" segment is accepted for ADMN_ALT_SETR) |
| 8 | 1 | Flag byte. Ignored. |
| 9 | 2 | Number of fields to update within a segment |
| 11 | * | First field for segment |

Each field entry is mapped as follows:

*Table 61. Field Entry Format*

| Offset | Length | Description |
|---|---|---|
| 0 | 8 | Field name as defined below. All field names must be left justified, entered in all uppercase, and padded with blanks |
| 8 | 1 | Field-specific flag byte |
| 9 | 2 | Length of field data |
| 11 | * | Field data |

The following rules apply:

- For boolean fields, the flag byte value of 'Y' or 'N' determines the flag setting. No field data is allowed for boolean fields. Coding field data results in an "Invalid Request" error being returned to the caller.
- For text fields, the flag byte 'Y' indicates the field should be created (or altered) with the specified field data. The flag byte 'N' indicated the field should be deleted (or not created). For flag byte 'N', any field data specified is ignored.
- For repeating text fields, a flag byte of 'A' indicates to add the specified field data to the existing data (if any). Field data must be specified, otherwise an "Invalid Request" error is returned to the caller. The flag byte 'D' indicates to delete the specified data (if any).

In addition to the flag bytes specified in the following tables, a flag byte of 'I' can be specified for any segment name. The callable service ignores any field data specified with a flag byte of 'I'.

The following table defines the SETROPTS field keywords and their usage. All field names relate directly to SETROPTS keywords. Refer to OS/390 Security Server (RACF) Command Language Reference for information about SETROPTS

keywords. Note that within the command image generated internally, RACF truncates long keywords to 12 characters.

*Table 62. BASE Segment Field Names*

| Field Name | Flag Byte Value | SETROPTS Keyword Reference |
|---|---|---|
| CLASSACT | 'A' | CLASSACT (xx ...) |
| | 'D' | NOCLASSACT (xx ...) |
| CLASSTAT | 'A' | STATISTICS (xx ...) |
| | 'D' | NOSTATISTICS (xx ...) |
| GENCMD | 'A' | GENCMD (xx ...) |
| | 'D' | NOGENCMD (xx ...) |
| GENERIC | 'A' | GENERIC (xx ...) |
| | 'D' | NOGENERIC (xx ...) |
| GENLIST | 'A' | GENLIST (xx ...) |
| | 'D' | NOGENLIST (xx ...) |
| GLOBAL | 'A' | GLOBAL (xx ...) |
| | 'D' | NOGLOBAL (xx ...) |
| RACLIST | 'A' | RACLIST (xx ...) |
| | 'D' | NORACLIST (xx ...) |
| INACTIVE | 'Y' | INACTIVE (xx ...) |
| | 'N' | NOINACTIVE (xx ...) |
| INITSTAT | 'Y' | INITSTATS (xx ...) |
| | 'N' | NOINITSTATS (xx ...) |
| TERMINAL | 'Y' | TERMINAL(xx) |
| AUDIT | 'A' | AUDIT (xx ...) |
| | 'D' | NOAUDIT (xx ...) |
| CMDVIOL (Boolean) | 'Y' | CMDVIOL |
| | 'N' | NOCMDVIOL |
| OPERAUDT (Boolean) | 'Y' | OPERAUDT |
| | 'N' | NOOPERAUDT |
| SAUDIT (Boolean) | 'Y' | SAUDIT |
| | 'N' | NOSAUDIT |
| APPLAUDT (Boolean) | 'Y' | APPLAUDIT |
| | 'N' | NOAPPLAUDIT |
| SLABAUDT (Boolean) | 'Y' | SECLABELAUDIT |
| | 'N' | NOSECLABELAUDIT |
| SLEVAUDT | 'Y' | SECLEVELAUDIT (xx ...) |
| | 'N' | NOSECLEVELAUDIT |
| LOGALWYS | 'Y' | LOGOPTIONS (ALWAYS (xx ...)) |
| LOGNEVER | 'Y' | LOGOPTIONS (NEVER (xx ...)) |
| LOGSUCC | 'Y' | LOGOPTIONS (SUCCESSES (xx ...)) |
| LOGFAIL | 'Y' | LOGOPTIONS (FAILURES (xx ...)) |

*Table 62. BASE Segment Field Names (continued)*

| Field Name | Flag Byte Value | SETROPTS Keyword Reference |
|---|---|---|
| LOGDEFLT | 'Y' | LOGOPTIONS (DEFAULT (xx ...)) |
| HISTORY | 'Y' | PASSWORD (HISTORY ( xx )) |
| | 'N' | PASSWORD (NOHISTORY) |
| INTERVAL | 'Y' | PASSWORD (INTERVAL ( xx )) |
| REVOKE | 'Y' | PASSWORD (REVOKE ( xx )) |
| | 'N' | PASSWORD (NOREVOKE) |
| WARNING | 'Y' | PASSWORD (WARNING ( xx )) |
| | 'N' | PASSWORD (NOWARNING) |
| RULES (Boolean) | 'N' | PASSWORD (NORULES) |
| NOTE: Specifying RULES with the 'N' flag results in the cancellation of all password syntax rules, regardless of any RULEn fields also specified. | | |
| RULE1 | 'Y' | PASSWORD (RULE1 (LENGTH (m1:m2) content-keyword (position))) |
| | 'N' | PASSWORD (NORULE1) |
| RULE2 | 'Y' | PASSWORD (RULE2 (LENGTH (m1:m2) content-keyword (position))) |
| | 'N' | PASSWORD (NORULE2) |
| RULE3 | 'Y' | PASSWORD (RULE3 (LENGTH (m1:m2) content-keyword (position))) |
| | 'N' | PASSWORD (NORULE3) |
| RULE4 | 'Y' | PASSWORD (RULE4 (LENGTH (m1:m2) content-keyword (position))) |
| | 'N' | PASSWORD (NORULE4) |
| RULE5 | 'Y' | PASSWORD (RULE5 (LENGTH (m1:m2) content-keyword (position))) |
| | 'N' | PASSWORD (NORULE5) |
| RULE6 | 'Y' | PASSWORD (RULE6 (LENGTH (m1:m2) content-keyword (position))) |
| | 'N' | PASSWORD (NORULE6) |
| RULE7 | 'Y' | PASSWORD (RULE7 (LENGTH (m1:m2) content-keyword (position))) |
| | 'N' | PASSWORD (NORULE7) |
| RULE8 | 'Y' | PASSWORD (RULE8 (LENGTH (m1:m2) content-keyword (position))) |
| | 'N' | PASSWORD (NORULE8) |

*Table 62. BASE Segment Field Names (continued)*

| Field Name | Flag Byte Value | SETROPTS Keyword Reference |
|---|---|---|
| NOTE: When specifying the 'Y' flag, the date supplied in the RULEn field consists of a length field and a character sequence, separated by a blank. The length field can be either a single numeric value, or two numeric values separated by a colon (:) to denote a minimum and maximum length. The character sequence conforms to the format of the output of the SETROPTS LIST command. It is a string of 1 to 8 characters, where each position of the string contains a character that indicates the valid characters that can occupy that position:<br>• A - Alphabetic<br>• C - Consonant<br>• L - Alphanumeric<br>• N - Numeric<br>• V - Vowel<br>• W - Non-vowel<br>• * - Any character<br><br>For example, if the RULE1 field is specified with field data of "3:6 A*NV*A", the resulting SETROPTS PASSWORD keyword would be RULE1(LENGTH(3:6) ALPHA(1 6) NUMERIC(3) VOWEL(4)).<br><br>See the *OS/390 SecureWay Security Server RACF Command Language Reference* for details on SETROPTS. | | |
| LIST (Boolean) | 'Y' | LIST |
| NOTE: The LIST field is not returned by ADMN_UNL_SETR or ADMN_XTR_SETR. | | |
| ADDCREAT (Boolean) | 'Y' | ADDCREATOR |
| | 'N' | NOADDCREATOR |
| ADSP (Boolean) | 'Y' | ADSP |
| | 'N' | NOADSP |
| CATDSNS | 'Y' | CATDSNS ( xx ) |
| | 'N' | NOCATDSNS |
| COMPMODE (Boolean) | 'Y' | COMPATMODE |
| | 'N' | NOCOMPATMODE |
| EGN (Boolean) | 'Y' | EGN |
| | 'N' | NOEGN |
| GENOWNER (Boolean) | 'Y' | GENERICOWNER |
| | 'N' | NOGENERICOWNER |
| GRPLIST (Boolean) | 'Y' | GRPLIST |
| | 'N' | NOGRPLIST |
| MLACTIVE | 'Y' | MLACTIVE ( xx ) |
| | 'N' | NOMLACTIVE |
| MLQUIET (Boolean) | 'Y' | MLQUIET |
| | 'N' | NOMLQUIET |
| MLS | 'Y' | MLS ( xx ) |
| | 'N' | NOMLS |

*Table 62. BASE Segment Field Names (continued)*

| Field Name | Flag Byte Value | SETROPTS Keyword Reference |
|---|---|---|
| MLSTABLE (Boolean) | 'Y' | MLSTABLE |
| | 'N' | NOMLSTABLE |
| PREFIX | 'Y' | PREFIX ( xx ) |
| | 'N' | NOPREFIX |
| PROTALL | 'Y' | PROTECTALL ( xx ) |
| | 'N' | NOPROTECTALL |
| REALDSN (Boolean) | 'Y' | REALDSN |
| | 'N' | NOREALDSN |
| REFRESH (Boolean) | 'Y' | REFRESH |
| NOTE: The REFRESH field is not returned by ADMN_UNL_SETR or ADMN_XTR_SETR. | | |
| RETPD | 'Y' | RETPD ( xx ) |
| RVARSWPW | 'Y' | RVARY ( SWITCH ( xx )) |
| NOTE: For ADMN_XTR_SETR, the value returned for this field is not the actual password, but one of two predefined values. A value of "DEFAULT" indicates that the default password in in effect, while a value of "INSTLN" indicates that an installation-defined password is in effect. | | |
| RVARSTPW | 'Y' | RVARY ( STATUS ( xx ) ) |
| NOTE: For ADMN_XTR_SETR, the value returned for this field is not the actual password, but one of two predefined values. A value of "DEFAULT" indicates that the default password in in effect, while a value of "INSTLN" indicates that an installation-defined password is in effect. | | |
| SECLABCT (Boolean) | 'Y' | SECLABELCONTROL |
| | 'N' | NOSECLABELCONTROL |
| SESSINT | 'Y' | SESSIONINTERVAL ( xx ) |
| | 'N' | NOSESSIONINTERVAL |
| TAPEDSN (Boolean) | 'Y' | TAPEDSN |
| | 'N' | NOTAPEDSN |
| WHENPROG (Boolean) | 'Y' | WHEN (PROGRAM) |
| | 'N' | NOWHEN (PROGRAM) |
| MODGDG (Boolean) | 'Y' | MODEL (GDG) |
| | 'N' | MODEL (NOGDG) |
| MODGROUP (Boolean) | 'Y' | MODEL (GROUP) |
| | 'N' | MODEL (NOGROUP) |
| MODUSER (Boolean) | 'Y' | MODEL (USER) |
| | 'N' | MODEL (NOUSER) |
| MODEL (Boolean) | 'N' | NOMODEL |
| ERASE (Boolean) | 'Y' | ERASE |
| | 'N' | NOERASE |
| ERASEALL (Boolean) | 'Y' | ERASE (ALL) |

*Table 62. BASE Segment Field Names (continued)*

| Field Name | Flag Byte Value | SETROPTS Keyword Reference |
|---|---|---|
| ERASESEC | 'Y' | ERASE (SECLEVEL ( xx ) ) |
| | 'N' | ERASE (NOSECLEVEL) |
| PRIMLANG | 'Y' | LANGUAGE (PRIMARY ( xx ) ) |
| SECLANG | 'Y' | LANGUAGE (SECONDARY ( xx ) ) |
| JESBATCH (Boolean) | 'Y' | JES (BATCHALLRACF) |
| | 'N' | JES (NOBATCHALLRACF) |
| JESEARLY (Boolean) | 'Y' | JES (EARLYVERIFY) |
| | 'N' | JES (NOEARLYVERIFY) |
| JESXBM (Boolean) | 'Y' | JES (XBMALLRACF) |
| | 'N' | JES (NOXBMALLRACF) |
| JESNJE | 'Y' | JES (NJEUSERID( xx ) ) |
| JESUNDEF | 'Y' | JES (UNDEFINEDUSER( xx ) ) |

*Output Message Block:* Following is the mapping of the output message block returned by R_admin for the ADMN_XTR_SETR and ADMN_UNL_SETR function codes. The output storage is obtained in the subpool specified by the caller in the Out_message_subpool parameter of IRRSEQ00.

*Table 63. Output Message Block*

| Offset | Length | Description |
|---|---|---|
| 0 | 4 | Eye catcher to aid in virtual storage dumps: 'RXTR" or 'RUNL' |
| 4 | 4 | Total length of output buffer |
| 8 | 4 | Reserved |
| 12 | 2 | Number of segment entries for ADMN_XTR_SETR, or number of record types returned for ADMN_UNL_SETR. |
| 12 | 2 | Number of RACF profile segments |
| 14 | 0 | Start of the first segment or record entry |

For ADMN_XTR_SETR, the output consists of a single segment entry for the base segment, followed by field entries for each of the supported input fields documented in "Parameter List Format for SETROPTS Administration" on page 91. Note that not all of the fields are returned, and fields that are not returned are noted in the field table. There is no defined order in which fields are returned. The segment and field entry for ADMN_XTR_SETR uses the standard ADMN_USRADM_SEGENTRY and ADMN_USRADM_FLDENTRY mappings used by other R_admin functions.

For ADMN_UNL_SETR, the output data is mapped using the following mapping for each unloaded record type, the number of which is contained in ADMN_EXTRACT_NUM. There is a single record type of "RACFINIT" to describe the basic RACF options. Following this record is a series of records of type "CLASNAME". There are as many "CLASNAME" records as there are classes defined in the class descriptor table (CDT). These include installation-defined

classes as well as classes supplied by IBM. Columns 44 through 51 of each record identify the name of the class that the record describes. See *OS/390 SecureWay Security Server RACF Macros and Interfaces* for detailed mappings of these record types. Note that for ADMN_UNL_SETR, R_admin does not fill in the time-written, date-written, and SMF system ID fields.

| Offset | Length | Description |
|---|---|---|
| 0 | 8 | The SMF data unload record type (documented in *OS/390 SecureWay Security Server RACF Macros and Interfaces*). |
| 8 | 4 | The length of an individual record of this type. |
| 12 | 4 | The number of records of this type. |
| 16 | 8 | Reserved |
| 24 | * | The start of the first record of this type. |

# R_audit (IRRSAU00): Provide an Audit Interface

## Function

The **R_audit** service provides an audit interface for functions that need to write an audit record for a condition where an audit by a security check service is not sufficient.

This service fills in the base part of the record and some standard relocate sections based on the function code in the CRED and the defined input parameters.

## Requirements

**Authorization:** Any PSW key in supervisor state
**Dispatchable unit mode:** Task of OS/390 UNIX user
**Cross memory mode:** PASN = HASN or PASN not = HASN
**AMODE:** 31
**RMODE:** Any
**ASC mode:** Primary or AR mode
**Recovery mode:** SETFRR
**Serialization:** Enabled for interrupts
**Locks:** No locks held
**Control parameters:** The parameter list and the work area must be in the primary address space. ALETs must be passed for all parameters except the work area. The words containing the ALETs must be in the primary address space.

## RACF Authorization

None

## Format

```
CALL IRRSAU00 (Work_area,
               ALET, SAF_return_code,
               ALET, RACF_return_code,
               ALET, RACF_reason_code,
               ALET, CRED,
               ALET, File_Identifier_1,
               ALET, FSP1,
               ALET, File_deleted_flag,
               ALET, File_Identifier_2,
               ALET, FSP2
              )
```

## Parameters

**Work_area**
The name of a 1024-byte work area for SAF and RACF usage. The work area must be in the primary address space.

**ALET**
The name of a word containing the ALET for the following parameter. Each parameter must have an ALET specified. Each ALET can be different. The words containing the ALETs must be in the primary address space.

**SAF_return_code**
The name of a fullword in which the SAF router returns the SAF return code.

**RACF_return_code**
The name of a fullword in which the service routine stores the return code.

**RACF_reason_code**
The name of a fullword in which the service routine stores the reason code.

**CRED**
The name of the CRED structure for the current file system syscall.

**File_Identifier_1**
The name of a 16-byte area containing a unique identifier of the file identified by the old (or only) pathname specified on the syscall.

**FSP1**
The name of the IFSP for the old (or only) file.

**File_deleted_flag**
For system calls that can cause a file to be deleted, the address of a byte containing a flag:

- 0 - the last link was not removed.
- 1 - the last link was removed for a file. The file is deleted.

**File_Identifier_2**
For system calls that create a new file name. If the “new” file existed, this is the name of a 16-byte area containing a unique identifier of the “new” file.

**FSP2**
The name of the IFSP for the new file.

## Return and Reason Codes

IRRSAU00 returns the following values in the reason and return code parameters:

| SAF Return Code | RACF Return Code | RACF Reason Code | Explanation |
|---|---|---|---|
| 0 | 0 | 0 | The service was successful. |
| 4 | 0 | 0 | RACF is not installed. |
| 8 | 8 | 24 | The audit function code is not valid. |
| 8 | 8 | 32 | The CRED user type is not supported. |

## Usage Notes

1. This service can be called by the MVS BCP, by an OS/390 UNIX System Services file system, or by OS/390 UNIX System Services servers. The service contains support for OS/390 UNIX System Services servers, but can not be directly invoked by an OS/390 UNIX System Services server.
2. IRRSAU00 tests whether auditing is required and, if so, builds and writes an audit record. The record built contains data from the calling process's security attributes (USP) and from the input CRED, the input IFSPs, and the input parameters. The content depends on the function being audited, as determined from the CRED_audit_function_code.
3. See *OS/390 SecureWay Security Server RACF Macros and Interfaces* for tables describing the data included in audit records, the data included in each event record, and syscalls that cause the event records to be written.

## Related Services

None

---

# R_chaudit (IRRSCA00): Change Audit Options

## Function

The **R_chaudit** service verifies that the user has authority to change the audit options for the specified file and, if so, sets the audit bits from the input audit options parameter.

## Requirements

**Authorization:** Any PSW key in supervisor state
**Dispatchable unit mode:** Task of OS/390 UNIX user
**Cross memory mode:** PASN = HASN or PASN not = HASN
**AMODE:** 31
**RMODE:** Any
**ASC mode:** Primary or AR mode
**Recovery mode:** SETFRR
**Serialization:** Enabled for interrupts
**Locks:** No locks held
**Control parameters:** The parameter list and the work area must be in the primary address space. ALETs must be passed for all parameters except the work area. The words containing the ALETs must be in the primary address space.

## RACF Authorization

None

## Format

```
CALL IRRSCA00 (Work_area,
               ALET, SAF_return_code,
               ALET, RACF_return_code,
               ALET, RACF_reason_code,
               ALET, Audit_options,
               ALET, FSP,
               ALET, File_identifier,
               ALET, CRED
              )
```

## Parameters

**Work_area**
The name of a 1024-byte work area for SAF and RACF usage. The work area must be in the primary address space.

**ALET**
The name of a word containing the ALET for the following parameter. Each parameter must have an ALET specified. Each ALET can be different. The words containing the ALETs must be in the primary address space.

**SAF_return_code**
The name of a fullword in which the SAF router returns the SAF return code.

**RACF_return_code**
The name of a fullword in which the service routine stores the return code.

**RACF_reason_code**
The name of a fullword in which the service routine stores the reason code.

**Audit_options**
The name of a word containing the audit options to be set. For RACF, the following options are defined:

- Audit options can be specified for each type of access:

  **Byte 1** read access audit options

  **Byte 2** write access audit options

  **Byte 3** execute/search access audit options

  **Byte 4** audit flag
- The following flags are defined for each of the first three bytes of audit options:

  **X'00'** do not audit any access attempts

  **X'01'** audit successful access attempts

  **X'02'** audit failed access attempts

  **X'03'** audit both types of attempts
- In the last byte (the audit flag), the last bit indicates whether user audit options or auditor audit options should be set:

  **X'00'** set user audit options

  **X'01'** set auditor audit options

Reserved bits in the audit options parameter must be zero.

**FSP**
The name of the IFSP for the file whose audit options are to be changed.

**File_Identifier**
The name of a 16-byte area containing a unique identifier of the file.

**CRED**
The name of the CRED structure for the current file system syscall. See *OS/390 SecureWay Security Server RACF Data Areas*.

## Return and Reason Codes

IRRSCA00 returns the following values in the reason and return code parameters:

| SAF Return Code | RACF Return Code | RACF Reason Code | Explanation |
|---|---|---|---|
| 0 | 0 | 0 | The service was successful. |
| 4 | 0 | 0 | RACF is not installed. |
| 8 | 8 | 4 | The user is not authorized to change the file's useraudit options. |
| 8 | 8 | 8 | The user is not authorized to change the file's audit or audit options. |
| 8 | 8 | 12 | An internal error occurred during RACF processing. |
| 8 | 8 | 24 | Reserved bits in an input parameter were not zero. |
| 8 | 8 | 32 | The CRED user type is not supported. |

## Usage Notes

1. This service is intended only for use by an OS/390 UNIX System Services file system and by OS/390 UNIX System Services servers. The service contains support for OS/390 UNIX System Services servers, but cannot be directly invoked by an OS/390 UNIX System Services server.
2. Two sets of audit bits exist for a file, one for auditor-specified options and one for user-specified options. The audit flag in the parameter list indicates which type of options should be set.

   If the audit flag indicates auditor options, the user must have auditor authority. Auditors can set the auditor options for any file, even those they do not have path access to or authority to use for any other reason.

   If the audit flag indicates user options, the user must be a superuser or must be the owner of the file (that is, the effective UID of the calling process is equal to the owner UID of the file.)
3. If the change is being made for an open file, that pathname in the CRED is not used.
4. An audit record is optionally written, depending on the audit options in effect for the system.

## Related Services

None

# R_chmod (IRRSCF00): Change File Mode

## Function

The **R_chmod** service checks whether the calling process is authorized to change the mode of the specified file (identified by the input IFSP) and, if so, changes the permission bits and the S_ISUID, S_ISGID, and S_ISVTX bits in the IFSP to the values specified by the mode parameter.

## Requirements

| | |
|---|---|
| **Authorization:** | Any PSW key in supervisor state |
| **Dispatchable unit mode:** | Task of OS/390 UNIX user |
| **Cross memory mode:** | PASN = HASN or PASN not = HASN |
| **AMODE:** | 31 |
| **RMODE:** | Any |
| **ASC mode:** | Primary or AR mode |
| **Recovery mode:** | SETFRR |
| **Serialization:** | Enabled for interrupts |
| **Locks:** | No locks held |
| **Control parameters:** | The parameter list and the work area must be in the primary address space. ALETs must be passed for all parameters except the work area. The words containing the ALETs must be in the primary address space. |

## RACF Authorization

1. To change the mode, the user must be a superuser or must be the owner of the file. If the user can change the mode and the user is not a superuser, the S_ISGID bit is cleared, except when the owner OS/390 UNIX group identifier (GID) of the file is equal to the effective GID or to one of the supplementary groups of the calling process.
2. Only a superuser or directory/file owner can change the S_ISVTX bit.

## Format

```
CALL IRRSCF00 (Work_area,
               ALET, SAF_return_code,
               ALET, RACF_return_code,
               ALET, RACF_reason_code,
               ALET, Mode,
               ALET, FSP,
               ALET, File_Identifier,
               ALET, CRED
              )
```

## Parameters

**Work_area**
The name of a 1024-byte work area for SAF and RACF usage. The work area must be in the primary address space.

**ALET**
The name of a word containing the ALET for the following parameter. Each parameter must have an ALET specified. Each ALET can be different. The words containing the ALETs must be in the primary address space.

**SAF_return_code**
The name of a full word in which the SAF router returns the SAF return code.

**RACF_return_code**
The name of a fullword in which the service routine stores the return code.

**RACF_reason_code**
The name of a fullword in which the service routine stores the reason code.

**Mode**
The name of a word containing the mode value (the file type, the permission bits, and the S_ISUID, S_ISGID, and S_ISVTX bits) to be set in the IFSP for the file.

See "File Type and File Mode Values" on page 4 for a definition of the security bits in the mode parameter. Reserved bits in the mode parameter must be zero.

**FSP**
The name of the IFSP for the file whose mode bits are to be changed.

**File_Identifier**
The name of a 16-byte area containing a unique identifier of the file.

**CRED**
The name of the CRED structure for the current file system syscall. See *OS/390 SecureWay Security Server RACF Data Areas*.

## Return and Reason Codes

IRRSCF00 returns the following values in the reason and return code parameters:

| SAF Return Code | RACF Return Code | RACF Reason Code | Explanation |
|---|---|---|---|
| 0 | 0 | 0 | The service was successful. |
| 4 | 0 | 0 | RACF is not installed. |
| 8 | 8 | 4 | The user is not authorized to change the mode of the file. |
| 8 | 8 | 12 | An internal error occurred during RACF processing. |
| 8 | 8 | 32 | The CRED user type is not supported. |

## Usage Notes

1. This service is intended only for use by an OS/390 UNIX System Services file system and by OS/390 UNIX System Services servers. The service contains support for OS/390 UNIX System Services servers, but cannot be directly invoked by an OS/390 UNIX System Services server.
2. The mode word is mapped by the OS/390 UNIX System Services macro BPXYMODE.
3. If the audit function code indicates an open file, the path name in the CRED is not used.
4. An audit record is optionally written, depending on the audit options in effect for the system.

## Related Services

makeFSP, R_umask

# R_chown (IRRSCO00): Change Owner and Group

## Function

The **R_chown** service checks to see whether the user is authorized to change the owner of the file, and, if so, changes the owner OS/390 UNIX user identifier (UID) and OS/390 UNIX group identifier (GID) to the specified values.

If the user is authorized to change the file, the S_ISUID and S_ISGID bits are cleared in the IFSP.

## Requirements

| | |
|---|---|
| **Authorization:** | Any PSW key in supervisor state |
| **Dispatchable unit mode:** | Task of OS/390 UNIX user |
| **Cross memory mode:** | PASN = HASN or PASN not = HASN |
| **AMODE:** | 31 |
| **RMODE:** | Any |
| **ASC mode:** | Primary or AR mode |
| **Recovery mode:** | SETFRR |
| **Serialization:** | Enabled for interrupts |
| **Locks:** | No locks held |
| **Control parameters:** | The parameter list and the work area must be in the primary address space. ALETs must be passed for all parameters except the work area. The words containing the ALETs must be in the primary address space. |

## RACF Authorization

1. This service implements the _POSIX_CHOWN_RESTRICTED feature in POSIX 1003.1.

   If _POSIX_CHOWN_RESTRICTED is in effect, then:

   - A user can change the owner OS/390 UNIX user identifier (UID) value only if the user is a superuser.
   - A user can change the owner OS/390 UNIX group identifier (GID) of a file if:
     - The user is a superuser,
     - Or, all of the following are true:
       - The effective UID of the calling process is equal to the owner UID of the file (that is, the user is the owner of the file).
       - The input UID is equal to the owner UID of the file or -1
       - The input OS/390 UNIX group identifier (GID) is equal to the effective GID or to one of the supplemental groups of the calling process.

   If _POSIX_CHOWN_RESTRICTED is not in effect, then:

   - A user can change the owner OS/390 UNIX user identifier (UID) value AND the owner OS/390 UNIX group identifier (GID) of a file if:
     - The user is a superuser
     - The effective UID of the calling process is equal to the owner UID of the file ( that is, the user is the owner of the file)

2. If the caller is not superuser, an authorization check is performed on the resource name in the UNIXPRIV class indicated in Table 64. If the authorization check is successful, the caller is treated as a superuser.

*Table 64. UNIXPRIV class resource names used in R_chown*

| Audit function code | Resource name | Access required |
|---|---|---|
| N/A | SUPERUSER.FILESYS.CHOWN | READ |

## Format

```
CALL IRRSCO00 (Work_area,
               ALET, SAF_return_code,
               ALET, RACF_return_code,
               ALET, RACF_reason_code,
               ALET, UID,
               ALET, GID,
               ALET, FSP,
               ALET, File_identifier,
               ALET, CRED
              )
```

## Parameters

**Work_area**
The name of a 1024-byte work area for SAF and RACF usage. The work area must be in the primary address space.

**ALET**
The name of a word containing the ALET for the following parameter. Each parameter must have an ALET specified. Each ALET can be different. The words containing the ALETs must be in the primary address space.

**SAF_return_code**
The name of a fullword in which the SAF router returns the SAF return code.

**RACF_return_code**
The name of a fullword in which the service routine stores the return code.

**RACF_reason_code**
The name of a fullword in which the service routine stores the reason code.

**UID**
The name of a word containing the OS/390 UNIX user identifier (UID) to be set as the file owner UID or -1 to indicate that:

1. This field is not changed in the IFSP.
2. The OS/390 UNIX group identifier (GID) can be changed.

**GID**
The name of a word containing the OS/390 UNIX group identifier (GID) to be set as the file owner GID or -1 to indicate that this field is not changed in the IFSP.

**FSP**
The name of the IFSP for the file whose owner OS/390 UNIX user identifier (UID) and OS/390 UNIX group identifier (GID) are to be changed.

**File_Identifier**
The name of a 16-byte area containing a unique identifier of the file.

**CRED**
The name of the CRED structure for the current file system syscall. See *OS/390 SecureWay Security Server RACF Data Areas*.

## Return and Reason Codes

IRRSCO00 returns the following values in the reason and return code parameters:

| SAF Return Code | RACF Return Code | RACF Reason Code | Explanation |
|---|---|---|---|
| 0 | 0 | 0 | The service was successful. |
| 4 | 0 | 0 | RACF is not installed. |
| 8 | 8 | 4 | The OS/390 UNIX user identifier (UID) is not valid. |
| 8 | 8 | 8 | The OS/390 UNIX group identifier (GID) is not valid. |
| 8 | 8 | 12 | An internal error occurred during RACF processing. |
| 8 | 8 | 20 | The user is not authorized to change the owner UID or GID. |
| 8 | 8 | 32 | The CRED user type is not supported. |
| 8 | 8 | 36 | The user is not authorized to set the specified GID. |

## Usage Notes

1. This service is intended only for use by an OS/390 UNIX System Services file system and by OS/390 UNIX System Services servers. The service contains support for OS/390 UNIX System Services servers, but can not be directly invoked by an OS/390 UNIX System Services server.
2. If the input UID or GID (or both) is equal to -1, that field is not changed in the IFSP.
3. If the audit function code indicates an open file, the pathname in the CRED is not used.
4. An audit record is optionally written, depending on the audit options in effect for the system.

## Related Services

query_file_security_options

# R_datalib (IRRSDL00): OCSF Data Library

## Function

The **R_datalib** service provides the function required to implement the Open Cryptographic Services Facility Data library functions.

## Requirements

**Authorization:** PSW key 8, non-APF authorized, problem state

**Dispatchable unit mode:** Task

| | |
|---|---|
| **Cross memory mode:** | PASN = HASN |
| **AMODE:** | 31 |
| **RMODE:** | Any |
| **ASC mode:** | Primary or AR mode |
| **Recovery mode:** | ESTAE. Caller cannot have an FRR active. |
| **Serialization:** | Enabled for interrupts |
| **Locks:** | No locks held |
| **Control parameters:** | The parameter list and the work area must be in the primary address space. The words containing the ALETs must be in the primary address space. |

## Linkage Conventions

The parameter list for this callable service is intended to be variable length to allow for future expansion. To allow for this, the last word in the parameter list must have a 1 in the high-order (sign) bit.

## RACF Authorization

- The following authority is required for the DataGetFirst, DataGetNext, and GetUpdateCode functions:
  1. If the RACF_user_ID field is the same as the user ID of the command issuer, then the required authority is READ to resource IRR.DIGTCERT.LISTRING in the FACILITY class.
  2. If the RACF_user_ID field is not the same as the user ID of the command issuer, then the required authority is UPDATE to the resource IRR.DIGTCERT.LISTRING in the FACILITY class.
- The CheckStatus function requires READ authority to the resource IRR.DIGTCERT.LIST in the FACILITY class.
- The DataAbortQuery function requires no authority.
- RACF SPECIAL or CONTROL authority to the resource IRR.DIGTCERT.GENCERT in the FACILITY class is required to retrieve the private keys of CERTAUTH and SITE certificates.
- The IncSerialNum function requires RACF SPECIAL or appropriate authority to the resource IRR.DIGTCERT.GENCERT in the FACILITY class, READ if the certificate is owned by the caller, CONTROL if the certificate is a SITE or CERTAUTH certificate.

## Format

```
CALL IRRSDL00 (Work_area,
               ALET, SAF_return_code,
               ALET, RACF_return_code,
               ALET, RACF_reason_code,
               Function_code,
               Attributes,
               RACF_user_ID,
               Ring_name,
               Parm_list_version,
               Parmlist
              )
```

## Parameters

**Work_area**
The name of a 1024-byte work area for SAF and RACF usage. The work area must be in the primary address space and must be on a double word boundary.

**ALET**
The name of a word containing the ALET for the following parameter. Each ALET can be different. The words containing the ALETs must be in the primary address space.

**SAF_return_code**
The name of a fullword in which the SAF router returns the SAF return code.

**RACF_return_code**
The name of a fullword in which the service routine stores the return code.

**RACF_reason_code**
The name of a fullword in which the service routine stores the reason code.

**Function_code**
The name of a 1 byte input area containing the function code

**X'01'** **DataGetFirst:** Locate and return the first trusted certificate in the ring specified in Ring_name, based on the selection criteria.

On a DataGetFirst function, the user may specify some selection criteria by setting Number_predicates to 1, and then supplying some attribute information, such as attribute type, and the length and address of the attribute data. The data in the returned certificate will match the attribute data supplied.

**X'02'** **DataGetNext:** Locate and return the next trusted certificate in the ring, based on the criteria specified in DataGetFirst.

**X'03'** **DataAbortQuery:** Free resources from previous DataGetFirst and DataGetNext requests.

**X'04'** **CheckStatus:** Return the TRUST/ NOTRUST status for a specified certificate.

**X'05'** **GetUpdateCode:** Return the sequence number for the ring specified. A change in the ring sequence number, (from a previously obtained ring sequence number,) indicates that the ring has changed. A ring is considered changed when the list of certificates in the ring has changed, or the digital certificate information for a certificate in the ring has changed.

**X'06'** **IncSerialNum:**Increment and return the last serial number field (CERTLSER) associated with the input certificate.

If the function code is not one of the preceding values, a parameter list error is returned.

**Attributes**
The name of a fullword area containing information about the function to be performed. To ensure compatibility with future enhancements, this field must be zero. IRRSDL00 ignores any attribute data.

**RACF_userid**
The name of a 9 byte input area that consists of a 1 byte length field followed

by up to 8 characters. It must be specified in upper case. If not specified, the length must equal 0. If not specified, the current user ID is the ring owner.

**Ring_name**
The name of a variable length input area that consists of a 1 byte length followed by up to 237 characters that represent the ring name. Ring_name is ignored for the CheckStatus, DataAbortQuery, and DataGetNext functions. Ring_name is used for the DataGetFirst service and the GetUpdateCode service.

**parm_list_version**
A four byte input value which contains the version number for the following field, parm_list. This field must be set to zero.

**Parm_list**
Specifies the address of the function specific parameter list for the function specified in Function_code. These are defined in "Function Specific Parameter Lists for IRRSDL00".

## Return and Reason Codes

IRRSDL00 may return one of several values as the SAF and RACF return and reason codes. This section defines the return codes which can be returned by any of the functions.

*Table 65. IRRSDL00 Return Codes*

| SAF Return Code | RACF Return Code | RACF Reason Code | Explanation |
|---|---|---|---|
| 0 | 0 | 0 | The service was successful. |
| 4 | 0 | 0 | RACF is not installed. |
| 8 | 8 | 4 | Parameter list error occurred. Attributes was not specified as 0 or the last word in the parameter list did not have the higher order bit on. |
| 8 | 8 | 8 | Not RACF authorized to use the requested service. |
| 8 | 8 | 12 | Internal error caused recovery to get control. |
| 8 | 8 | 16 | Unable to establish a recovery environment. |
| 8 | 8 | 20 | Requested Function_code not defined. |
| 8 | 8 | 24 | Parm_list_version number not supported. |
| 8 | 8 | 28 | Error in Ring_name length or RACF_userid length. |

# Function Specific Parameter Lists for IRRSDL00

For each function code, there is a function specific parameter list.

## Function Specific Parameter Lists for DataGetFirst and DataGetNext

**Results_handle**
A 4 byte address pointing to an input area. The input area must be at least 20 bytes in length, and it is that input area (not Results_handle) that is mapped as follows:

**dbToken**
A 4 byte value reserved for use by RACF. This value must be preserved for subsequent calls to DataGetNext and DataAbortQuery.

**Number_predicates**
A 4 byte integer input value. A zero,X'00000000', indicates that there are no selection criteria. This value is only used on a DataGetFirst function. A one, X'00000001', indicates that a query on a particular attribute is being performed. All other values result in a parameter error.

**Attribute_ID**
A 4 byte integer input value which identifies the attribute that is being queried. This field is ignored if Number_predicates is zero, X'00000000'. If Number_predicates is one, X'00000001' this field must have one of the values listed below:

**X'00000001'**
Attribute data to match on is label.

**X'00000002'**
Attribute data to match on is default flag. Attribute data supplied by Attribute_length or Attribute_ptr will either be zero, X'00000000' or non-zero.

**X'00000003'**
Attribute data to match on is the DER encoded subject's distinguished name.

If the Attribute_ID is not one of the preceding values, a parameter ist error is returned.

**Attribute_length**
A 4 byte input value containing the length of the attribute data. This field is ignored if Number_predicates is zero, X'00000000'.

**Attribute_ptr**
A 4 byte input address which points to the attribute data. This field is ignored if Number_predicates is zero, X'00000000'.

**Certificate_Usage**
A 32 bit output flag value, indicating the usage of the certificate. Certificate_Usage may have the values:

**X'00000002'**
Certauth

**X'00000008'**
Personal

**X'00000000'**
Other (site)

**X'fffffff5'**
Reserved

**Default**
A 4 byte output value. A X'00000000' value indicates that this certificate is not the default certificate for the ring. A non-zero value indicates that this is the default certificate for the ring.

**Certificate_length**
A 4 byte value containing the length of the certificate. On input, it contains the length of the field pointed to by certificate_ptr. On output, it contains the actual size of the certificate that is returned. A zero indicates that no certificate was returned.

**Certificate_ptr**
A 4 byte input value containing the address of the DER encoded certificate output area.

**Private_key_length**
A 4 byte value containing the length of the private key. On input, it contains the length of the field pointed to by private_key_ptr. On output, it contains the actual size of the private key that is returned. A zero indicates that no private_key was returned.

If private_key_length is zero, then private_key_bitsize and private_key_type are not returned.

**Private_key_ptr**
A 4 byte input value containing the address of the private key output area.

**Private_key_type**
A 4 byte output value indicating the form of the private key. The valid values are:

**X'00000001'**
PKCS #1 private key, DER encoded

**X'00000002'**
ICSF key token lsbel

**Private_key_bitsize**
A 4 byte output value indicating the size of the private key, expressed in bits.

**Label_length**
A 4 byte value containing the length of the label. On input, it contains the length of the field pointed to by label_ptr. This field must be at least 32 bytes long. On output, it contains the actual size of the label that is returned. A zero value indicates that no label was returned.

**Label_ptr**
A 4 byte input value containing the address of the label output area.

**CERT_user_ID**
A 9 byte output area containing a 1 byte length, followed by the user ID which owns the certificate.

The 1 byte length must specify a length of 8. The user ID must be left-justified and padded with blanks.

**Subjects_DN_length**
A 4 byte input value containing the length of the DER encoded subject's distinguished name. On input, it contains the length of the field pointed to by Subjects_DN_ptr. On output, it contains the actual size of the subject's distinguished name that is returned.

**Subjects_DN_ptr**
a 4 byte input value containing the address of the subject's distinguished name output area.

**Record_ID_length**
On input, it contains the length of the field pointed to by Record_ID_ptr. This field must have a length of at least 246 bytes. On output, it contains the actual size of the record ID that is returned. A zero value indicates that no record ID was returned.

**Record_ID_ptr**
A 4 byte input value which points to a caller-provided 246 byte output area. This output area contains the record_ID returned from the callable service.

## Function Specific Parameter Lists for IncSerialNum

**Certificate_length**
The 4 byte input value, containing the length of the certificate.

**Certificate_ptr**
The 4 byte input address value, containing the address of the DER encoded certificate.

**Serial_number**
The 8 byte output area to contain the returned serial number.

## Usage Notes

1. A private key is only returned when:
   - The certificate's ring usage is personal, and
   - the caller's user ID is the user ID associated with the certificate profile or, for CERTAUTH and SITE certificates, the caller is RACF SPECIAL or has CONTROL authority to the resource IRR.DIGTCERT.GENCERT in the FACILITY class.
2. The DataAbortQuery function must be called once for each DataGetFirst call, whether or not DataGetNext calls are made between the DataGetFirst and DataAbortQuery calls. The caller must pass the same dbToken to DataAbortQuery call as was returned from the DataGetFirst call. If these conditions are not met, system resources will not be freed.

## Return and Reason Codes

The return codes that may be returned from the DataGetFirst and DataGetNext functions are:

*Table 66. DataGetFirst and DataGetNext Return Codes*

| SAF Return Code | RACF Return Code | RACF Reason Code | Explanation |
|---|---|---|---|
| 8 | 8 | 32 | Length error in attribute_length, Record_ID_length, label_length, or CERT_user_ID. |
| 8 | 8 | 36 | dbToken error. The token may be zero, in use by another task, or may have been created by another task. |
| 8 | 8 | 40 | Internal error while validating dbToken. |
| 8 | 8 | 44 | Record not found. |

*Table 66. DataGetFirst and DataGetNext Return Codes (continued)*

| SAF Return Code | RACF Return Code | RACF Reason Code | Explanation |
|---|---|---|---|
| 8 | 8 | 48 | An output area is not long enough. One or more of the following input length fields were too small: Certificate_length, Private_key_length, or Subjects_DN_length. The length field(s) returned contain the amount of storage needed for the service to successfully return data. |
| 8 | 8 | 52 | Internal error while obtaining record private key data. |
| 8 | 8 | 56 | Parameter error - Number_predicates or Attribute_ID in error. |
| 8 | 8 | 80 | Internal error while obtaining ring certificate information or record trust information. |
| 8 | 8 | 84 | Profile for Ring_name not found. |

The return codes that may be returned from the IncSerialNum functions are:

*Table 67. IncSerialNum Codes*

| SAF Return Code | RACF Return Code | RACF Reason Code | Explanation |
|---|---|---|---|
| 0 | 0 | 0 | Success, serial number returned. |
| 8 | 8 | 76 | Certificate is invalid. |
| 8 | 8 | 80 | Certificate is not installed or is marked NOTRUST. |
| 8 | 8 | 84 | Parameter error. Zero value specified for certificate length or certificate owned by another user. |

## Function Specific Parameter List for the DataAbortQuery Function

**Results_handle**
The 4 byte address value, which was returned from a prior DataGetFirst or DataGetNext request. This value is provided by the caller. The data area pointed to by Results_handle must have its fields preserved from prior DataGetFirst and DataGetNext calls.

Note that a DataAbortQuery call is required regardless of the return and reason codes from the corresponding DataGetFirst call. The DataAbortQuery function requires no authority.

**Return and Reason Codes:** The return codes that may be returned from the DataAbortQuery function are:

*Table 68. DataAbortQuery Return Codes*

| SAF Return Code | RACF Return Code | RACF Reason Code | Explanation |
|---|---|---|---|
| 8 | 8 | 36 | dbToken error. The token may be zero, in use by another task, or may have been created by another task. |
| 8 | 8 | 40 | Internal error while validating dbToken. |

## Function Specific Parameter List for the CheckStatus Function

**Certificate_length**
The 4 byte input value, containing the length of the certificate.

**Certificate_ptr**
The 4 byte input address value, containing the address of the DER encoded certificate.

**CheckStatus Return Codes:** The return codes that may be returned from the CheckStatus function are:

*Table 69. CheckStatus Return Codes*

| SAF Return Code | RACF Return Code | RACF Reason Code | Explanation |
|---|---|---|---|
| 0 | 0 | 0 | Certificate is trusted or certificate not registered with RACF. |
| 8 | 8 | 60 | Internal error - Unable to decode certificate. |
| 8 | 8 | 64 | Certificate is registered with RACF as not trusted. |
| 8 | 8 | 68 | Parameter error - zero value specified for Certificate_length or Certificate_ptr. |

## Function Specific Parameter List for GetUpdateCode

**Ring_sequence_number**
A four byte output field containing the sequence number for the ring specified.

**GetUpdateCode Return Codes:** The return codes that may be returned from the GetUpdateCode function are:

*Table 70. GetUpdateCode Return Codes*

| SAF Return Code | RACF Return Code | RACF Reason Code | Explanation |
|---|---|---|---|
| 8 | 8 | 84 | Profile for Ring_name not found. |
| 8 | 8 | 88 | Internal error - Unable to obtain ring data. |

**Related Services:** None

# R_dceauth (IRRSDA00): Check a User's Authority

## Function

The R_dceauth service enables an application server to check a RACF-defined user's authority to access a RACF-defined resource. It is intended to be used only by the OS/390 UNIX kernel on behalf of an application server.

The client's identity can be specified by:

- The ACEE
- The *RACF_userid* parameter
- The cell and principal UUID parameter pair

## Requirements

| | |
|---|---|
| **Authorization:** | Any PSW key in supervisor state |
| **Dispatchable unit mode:** | Task of OS/390 UNIX user |
| **Cross memory mode:** | PASN = HASN |
| **AMODE:** | 31 |
| **RMODE:** | Any |
| **ASC mode:** | Primary or AR ASC mode |
| **Recovery mode:** | ESTAE. Caller cannot have an FRR active. |
| **Serialization:** | Enabled for interrupts |
| **Locks:** | No locks held |

## RACF Authorization

1. RACF checks the ACEE, *RACF_userid*, and the UUID parameters in the following order:
   - If the ACEE parameter has been specified, this parameter is used to identity the user for this authorization request.
   - If the ACEE parameter has not been specified, and the *RACF_userid* parameter is present, this parameter is used to identify the user for this authorization request.
   - If neither the ACEE nor the *RACF_userid* parameter is present, the *Cell_string_uuid* and the *Principal_string_uuid* parameters are used to identify the user for this authorization request.
   - If the ACEE, *RACF_userid*, and UUID parameters have not been supplied, RACF uses the current task level ACEE if it is found. If there is no task level ACEE, RACF uses the address space ACEE, if it is present, to identify the user for this authorization request.

## Format

```
CALL IRRSDA00 (Work_area,
               ALET, SAF_return_code,
               ALET, RACF_return_code,
               ALET, RACF_reason_code,
                     ACEE_ptr,
               ALET,(Principal_string_uuid,
                     Cell_string_uuid,
                     RACF_userid,
                     RACF_class,
                     entity_name,
                     entity_length
                     access_requested)
              )
```

## Parameters

**Work_area**
The name of a 1024-byte work area for SAF and RACF usage. The work area must be in the primary address space.

**ALET**
The name of a word containing the ALET for the following parameter. Each ALET can be different. The words containing the ALETs must be in the primary address space.

**SAF_return_code**
The name of a fullword in which the SAF router returns the SAF return code.

**RACF_return_code**
The name of a fullword in which the service routine stores the return code.

**RACF_reason_code**
The name of a fullword in which the service routine stores the reason code.

**ACEE_ptr**
The name of a fullword containing the address of an area that contains a previously created ACEE. If the caller does not specify an ACEE, this area must contain binary zeros. The ACEE parameter is **not** specified by an ALET. This parameter must be in the primary address space.

**ALET**
The name of a word which must be in the primary address space and which contains the ALET for the following fields:
- Principal_string_uuid
- Cell_string_uuid
- RACF_userid
- RACF_class
- Entity_name
- Entity_length
- Access_requested

**Principal_string_uuid**
The name of an area containing the string form of the client's DCE UUID. If the caller does not specify the client's DCE UUID, then the first character of the area must be a NULL byte. (That is, the first byte of the 36-byte area must contain X'00'.)

**Cell_string_uuid**
The name of an area containing the string form of the cell DCE UUID. The string form of the cell UUID, if supplied by the caller, must be 36 bytes long.

The *Cell_string_uuid* must be the name of a 36-byte area that contains one of the following:

- The string form of the cell UUID
- A null byte (X'00') as the first character of the 36-byte area. If the home cell UUID is not applicable and the caller wants to obtain cross linking information using only the DCE principal UUID, the caller must pass the *Cell_string_uuid* parameter with the first byte of this field containing a null byte of X'00'.

**RACF_userid**
The name of a nine-byte area, which consists of a one-byte length field followed by up to eight characters. It must be specified in uppercase. If not specified, the length must equal zero.

**RACF_class**
The name of an eight-byte area containing the RACF class name of the resource (such as TAPEVOL). The class name must be

- Left justified
- Padded to the right with blanks
- Specified in uppercase
- Defined to RACF via the RACF class descriptor table supplied by IBM or the installation-defined RACF class descriptor table (described in the *OS/390 SecureWay Security Server RACF System Programmer's Guide*).

**Entity_name**
The name of an area that contains the RACF resource profile name (such as TAPE01). It must be specified in uppercase.

**Entity_length**
The name of an area that contains the halfword length of the entity_name. The valid range of this parameter is 1 to 246 characters.

**Access_requested**
The requested access (READ, UPDATE, CONTROL, ALTER) to the resource, which is the name of a one-byte area containing:

| Requested Access | Value |
|---|---|
| READ Access | X'02' |
| UPDATE Access | X'04' |
| CONTROL Access | X'08' |
| ALTER Access | X'80' |

## Return and Reason Codes

IRRSDA00 may return the following values in the reason and return code parameters:

| SAF return code | RACF return code | RACF reason code | Explanation |
|---|---|---|---|
| 0 | 0 | 0 | Access granted |
| 4 | 0 | 0 | RACF not installed, or RACF is not active |
| 8 | 8 | 4 | The resource is not defined to RACF |
| 8 | 8 | 8 | User is not authorized to access the resource |

| SAF return code | RACF return code | RACF reason code | Explanation |
|---|---|---|---|
| 8 | 8 | 12 | Internal processing error |
| 8 | 8 | 16 | Recovery environment could not be established |
| 8 | 8 | 20 | No mapping to a RACF user ID exists for the supplied UUID pair |
| 8 | 8 | 24 | Parameter list error |
| 8 | 8 | 28 | DCEUUIDS class is not active. RACF is not able to map the supplied UUIDs to a RACF user ID. |
| 8 | 8 | 32 | RACF was unable to create a security environment for the user ID specified. |
| 8 | 8 | 36 | The user ID is not RACF defined. |

## Usage Notes

1. This service may **not** be used to determine access to OS/390 UNIX resources, such as HFS files or to data sets.
2. The DATASET class is not valid for this service.

Parameter Usage:

| Parameter | Direction | Value |
|---|---|---|
| SAF_return_code | Output | |
| RACF_return_code | Output | |
| RACF_reason_code | Output | |
| ACEE_ptr | Input | Optional |
| Principal_string_uuid | Input | Optional |
| Cell_string_uuid | Input | Optional |
| RACF_userid | Input | Optional |
| RACF_class | Input | Required |
| entity_name | Input | Required |
| entity_length | Input | Required |
| access_requested | Input | Required |

## Related Services

R_dceruid

# R_dceinfo (IRRSDI00): Retrieve or Set User Fields

## Function

The RACF R_dceinfo callable service retrieves or sets the following fields from a user profile **DCE** segment:

- The DCE principal name associated with this RACF user
- DCE UUID of this user
- DCE cell name that this user is defined to (HOME CELL)
- DCE cell UUID that is associated with DCE cell that this user is defined to (HOMEUUID)

- A flag byte that indicates whether OS/390 DCE creates a DCE security context (autologin) automatically

The action performed by this callable service is based on the function code passed by the caller in the R_dceinfo parameter list.

- When the function code is set to EXTRACT, R_dceinfo retrieves the information requested from the user's DCE segment.
- When the function code is set to REPLACE, R_dceinfo replaces the fields that have been specified in the parameter list.

## Requirements

| | |
|---|---|
| **Authorization:** | Any PSW key in: Supervisor state for REPLACE DCE fields Supervisor or problem state for EXTRACT DCE fields |
| **Dispatchable unit mode:** | Task of OS/390 UNIX user |
| **Cross memory mode:** | PASN = HASN |
| **AMODE:** | 31 |
| **RMODE:** | Any |
| **ASC mode:** | Primary or AR mode |
| **Recovery mode:** | ESTAE. Caller cannot have an FRR active. |
| **Serialization:** | Enabled for interrupts |
| **Locks:** | No locks held |
| **Control parameters:** | The parameter list and the work area must be in the primary address space. ALETs must be passed for all parameters except the work area. The words containing the ALETs must be in the primary address space. |

## RACF Authorization

1. The replace function fails if the user ID specified in the parameter has not been previously defined as a DCE RACF user.
2. Field level access checking does not occur when retrieving or replacing fields with this service.

## Format

```
CALL IRRSDI00 (Work_area,
               ALET, SAF_return_code,
               ALET, RACF_return_code,
               ALET, RACF_reason_code,
               ALET, Function_code,
               ALET, RACF_userid,
               ALET, Field_list
               ALET, Output_area
               ALET, Output_area_length
              )
```

## Parameters

**Work_area**
The name of a 1024-byte work area for SAF and RACF usage. The work area must be in the primary address space.

**ALET**
The name of a word containing the ALET for the following parameter. Each

parameter must have an ALET specified. Each ALET can be different. The words containing the ALETs must be in the primary address space.

**SAF_return_code**
The name of a full word in which the SAF router returns the SAF return code.

**RACF_return_code**
The name of a fullword in which the service routine stores the return code.

**RACF_reason_code**
The name of a fullword in which the service routine stores the reason code.

**Function_code**
The name of a one-byte area containing the function code:

**X'01'** Retrieve DCE fields

**X'02'** Replace DCE fields

**RACF_userid**
The name of a nine-byte area, containing a one byte length field, followed by a user ID up to eight characters long. It must be specified in uppercase.

**Field_list**
The name of an area containing the fields to be replaced or retrieved. The format of the parameter list is:

| *Offset* | *Length* | *Description* |
|---|---|---|
| 0 | 2 | The length in bytes of the DCE field list |
| 2 | 2 | Total number of fields in the DCE field list |
| 4 | 8 | The name of the field |
| 12 | 2 | The length of the field data |
| 14 | variable | Field data |

The ordered triplet (name of the field, length of the field, and field data) is a repeating data structure. This triplet can repeat for the total number of fields in the input Field_list.

- If the function code is **X'01'** (**retrieve DCE fields**), the caller is expected to provide the following fields in the **Field_list**:
  - The total length in bytes of the input field list
  - The total number of fields to be retrieved
  - The name of the fields to be retrieved
  - The length of the data

  **Note:**

  For the retrieve function, there is no input field data. The caller is expected to provide a length of binary zero.

  The name of the field and the place holder of zero may repeat for the total number of fields to be retrieved.
- If the function code is **X'02'** (**replace DCE fields**), the caller is expected to be in supervisor state and to provide the following fields in the **Field_list**:
  - The total length in bytes of the input field list
  - The total number of fields to be retrieved
  - The name of the field to be retrieved
  - The length of the data

– Field data

A problem-state caller is not authorized to replace DCE information.

**Output_area**
The name of a fullword that contains the fields obtained by the R_dceinfo service when the function code is **X'01'** (**Retrieve DCE fields**). The format of the output area is:

| *Offset* | *Length* | *Description* |
|---|---|---|
| 0 | 2 | Total length of the output area of the retrieved data |
| 2 | 2 | SUBPOOL of the output area |
| 2 | 2 | Number of fields retrieved |
| 4 | 8 | Name of the retrieved field |
| 12 | 2 | Length of the retrieved field |
| 14 | variable | Field data |

The ordered triplet (name of the field, length of the field, and field data) is a repeating data structure. This triplet can repeat for the number of times shown in the output_area 'number of fields retrieved' count.

**Output_area_length**
The name of a fullword that contains the length of the output_area that is supplied by the caller of this service.

## Return and Reason Codes

IRRSDI00 may return the following values in the reason and return code parameters:

| **SAF return code** | **RACF return code** | **RACF reason code** | **Explanation** |
|---|---|---|---|
| 0 | 0 | 0 | Service successful. |
| 4 | 0 | 0 | RACF not installed. |
| 8 | 8 | 4 | Caller is not authorized. |
| 8 | 8 | 12 | Internal error during RACF processing. |
| 8 | 8 | 16 | Recovery Environment could not be established. |
| 8 | 8 | 20 | User does not have a DCE segment. |
| 8 | 8 | 24 | Length of the output area is too small to contain the data retrieved. |
| 8 | 8 | 28 | Parameter list error. |
| 8 | 8 | 32 | User ID specified does not exist. |

## Usage Notes

1. If the caller is in problem state, the *RACF_userid* specified must be the same RACF user as found in either the task level ACEE or the address space level ACEE.
2. If the caller is in supervisor state, the task and address space ACEEs are not checked. Therefore, an authorized caller may extract or replace DCE segment fields for any user who has a DCE segment.

3. The retrieve function returns fields that have been previously populated. Associated with the returned fields is a length indicator. The length indicator is set to zero if a field does not exist.
4. It is the responsibility of the caller to obtain and free the output area. If the fields to be retrieved from RACF are larger than the output area, RACF fails the request and returns the actual length required in the *output_area_length* parameter.
5. The field names supplied by the caller may be:
   - UUID
   - DCENAME
   - HOMECELL
   - HOMEUUID
   - DCEFLAGS

   The field names must be supplied as 8-character fields, left justified, and padded with blanks. They must be specified in uppercase.
6. The DCEFLAGS field is a one-byte field with the following meaning:
   - Value of X'00' means that OS/390 DCE does *not* attempt to sign on this user to OS/390 DCE automatically
   - Value of X'01' means that OS/390 DCE attempts to automatically sign on this user to OS/390 DCE

Parameter Usage:

| Parameter | Direction | |
|---|---|---|
| | **GET_INFO** | **PUT_INFO** |
| SAF_return_code | Output | Output |
| RACF_return_code | Output | Output |
| RACF_reason_code | Output | Output |
| Function_code | Input | Input |
| RACF_userid | Input | Input |
| Field_list | Input | Input |
| Output_area | Output | n/a |
| Output_area_length | Output | n/a |

## Related Services

R_dcekey

# R_dcekey (IRRSDK00): Retrieve or Set a DCE Password

## Function

The RACF R_dcekey callable service enables OS/390 DCE to retrieve or set a DCE password (*key*).

- This service retrieves the DCE password from a DCE segment. The password is decrypted using the key that was stored in the user's DCE segment when the password was encrypted.
- This service sets the DCE password in a user profile DCE segment, The password is encrypted using the key stored in the DCE.PASSWORD.KEY profile in the RACF KEYSMSTR general resource class.

The operation of **R_dcekey** is based on the function code values of **get_key** and **put_key** in the parameter list. See "Usage Notes" on page 126 for detailed information.

## Requirements

| | |
|---|---|
| **Authorization:** | Any PSW key in supervisor state |
| **Dispatchable unit mode:** | Task of OS/390 UNIX user |
| **Cross memory mode:** | PASN = HASN |
| **AMODE:** | 31 |
| **RMODE:** | Any |
| **ASC mode:** | Primary or AR mode |
| **Recovery mode:** | ESTAE. Caller cannot have an FRR active. |
| **Serialization:** | Enabled for interrupts |
| **Locks:** | No locks held |
| **Control parameters:** | The parameter list and the work area must be in the primary address space. ALETs must be passed for all parameters except the work area. The words containing the ALETs must be in the primary address space. |

## RACF Authorization

1. For this service to successfully complete, the user ID specified in the parameter must have a DCE segment.

## Format

```
CALL IRRSDK00 (Work_area,
               ALET, SAF_return_code,
               ALET, RACF_return_code,
               ALET, RACF_reason_code,
               ALET, Function_code,
               ALET, RACF_userid,
               ALET, key_area,
               ALET, key_area_length
              )
```

## Parameters

**Work_area**
The name of a 1024-byte work area for SAF and RACF use. The work area must be in the primary address space.

**ALET**
The name of a word containing the ALET for the following parameter. Each parameter must have an ALET specified. Each ALET can be different. The words containing the ALETs must be in the primary address space.

**SAF_return_code**
The name of a fullword in which the SAF router returns the SAF return code.

**RACF_return_code**
The name of a fullword in which the service routine stores the return code.

**RACF_reason_code**
The name of a fullword in which the service routine stores the reason code.

**Function_code**
The name of a one-byte area containing the function code:

**X'01'** get_key (retrieve current DCE key)

**X'02'** put_key (set DCE key)

**RACF_userid**
The name of a nine-byte area, which consists of a one-byte length field followed by up to eight characters. It must be specified in uppercase.

**Key_area**
The name of an area containing the DCE key, preceded by a two-byte length field.

**Key_area_length**
The name of a fullword that contains the length of the key_area.

## Return and Reason Codes

IRRSDK00 may return the following values in the reason and return code parameters:

| SAF return code | RACF return code | RACF reason code | Explanation |
|---|---|---|---|
| 0 | 0 | 0 | Service successful. |
| 4 | 0 | 0 | RACF not installed. |
| 8 | 8 | 4 | User is not defined to RACF. |
| 8 | 8 | 12 | Internal error during RACF processing. |
| 8 | 8 | 16 | Recovery environment could not be established. |
| 8 | 8 | 20 | User does not have a DCE segment. |
| 8 | 8 | 24 | A DCE Key has not been set for this user. |
| 8 | 8 | 28 | The key_area supplied by the caller is too small. |
| 8 | 8 | 32 | Parameter list error. |
| 8 | 8 | 36 | RACF KEYSMSTR inactive, or the profile DCE.PASSWORD.KEY profile was not defined to the RACF KEYSMSTR with a SSIGNON segment. |
| 8 | 8 | 40 | Invalid data was found in SSIGNON segment of the DCE.PASSWORD.KEY profile in the RACF KEYSMSTR. |
| 8 | 8 | 44 | RACF was unable to retrieve or update the master key/master key token in the SSIGNON segment of the DCE.PASSWORD.KEY profile in the RACF KEYSMSTR. |
| 8 | 8 | 48 | RACF was unable to retrieve or update the user's DCE key. |
| 8 | 8 | 52 | RACF cannot locate the CCA support routine. |
| 8 | 8 | 56 | The invocation of the CCA support routine has failed. |

## Usage Notes

1. When the function is **get_key**, this service returns the current DCE key in clear text form to the output area supplied by the caller. This is the value defined by the *key_area* parameter. The length of the *key_area* is supplied by the *key_area_length* parameter.
2. If the *key_area* supplied by the caller is too small to contain the user's current DCE key, the service sets the required length in the *key_area_length* parameter.
3. When the function is **put_key**, this service replaces the current DCE key in the specified user profile DCE segment with the value specified in the *key_area* parameter.

Parameter Usage:

| Parameter | Direction | |
|---|---|---|
| | **GET_KEY** | **PUT_KEY** |
| SAF_return_code | Output | Output |
| RACF_return_code | Output | Output |
| RACF_reason_code | Output | Output |
| Function_code | Input | Input |
| RACF_userid | Input | Input |
| Key_area | Output | Input |
| Key_area_length | Input/Output | n/a |

## Related Services

R_dceinfo

# R_dceruid (IRRSUD00): Determine the ID of a Client

## Function

The **R_dceruid** service enables OS/390 DCE servers to determine the RACF user ID of the client from the string forms of the client's DCE UUID pair, which consists of the *home cell* UUID and the *principal* UUID. It also enables the servers to determine the DCE UUIDs of a client from the RACF user ID.

Note that this service can *only* convert a DCE UUID to a RACF user ID and a RACF user ID to a DCE UUID for users who have:

- A populated DCE segment associated with their user profile
- A DCEUUIDS-class profile that defines the association between the DCE UUIDs and the RACF user ID

The R_dceruid service is sensitive to the profiles defined to the RACF DCEUUIDS class.

To resolve a conversion request:

- If a caller specifies a UUID pair on this service's invocation to convert the UUID pair to the corresponding RACF user ID, the service searches the RACF DCEUUIDS-class profiles defined to RACF with that UUID pair.

- If a caller specifies only a principal UUID on the service's invocation to convert the principal UUID to the corresponding RACF user ID, the service searches the RACF DCEUUIDS-class profiles defined to RACF with only that principal UUID.

## Requirements

| | |
|---|---|
| **Authorization:** | Any PSW key in supervisor or problem state |
| **Dispatchable unit mode:** | Task of OS/390 UNIX user |
| **Cross memory mode:** | PASN = HASN |
| **AMODE:** | 31 |
| **RMODE:** | Any |
| **ASC mode:** | Primary or AR mode |
| **Recovery mode:** | ESTAE. Caller cannot have a FRR active. |
| **Serialization:** | Enabled for interrupts |
| **Locks:** | No locks held |
| **Control parameters:** | The parameter list and the work area must be in the primary address space. ALETs must be passed for all parameters except the work area. The words containing the ALETs must be in the primary address space. |

## RACF Authorization

1. This service can only translate a RACF user ID to a DCE UUID and a DCE UUID to a RACF user ID for users who have:
   - A populated DCE segment associated with the user profile
   - A DCEUUIDS-class profile defined to RACF that associates a DCE UUID pair with a RACF user ID
2. Use of the R_dceruid service is authorized by the profile IRR.RDCERUID in the RACF FACILITY class. The user ID of the application server (the RACF identity associated with the application server), or a group to which the server is connected, must be permitted with READ access to the profile IRR.RDCERUID in the RACF FACILITY class. Assigning a UACC of READ to the profile IRR.RDCERUID is not recommended.

## Format

```
CALL IRRSUD00 (Work_area,
               ALET,SAF_return_code,
               ALET,RACF_return_code,
               ALET,RACF_reason_code,
               ALET,Function_code,
               ALET,Principal_string_uuid,
               ALET,Cell_string_uuid,
               ALET,RACF_userid
              )
```

## Parameters

**Work_area**
The name of a 1024-byte work area for SAF and RACF usage. The work area must be in the primary address space.

**ALET**
The name of a word containing the ALET for the following parameter. Each

parameter must have an ALET specified. Each ALET can be different. The words containing the ALETs must be in the primary address space.

**SAF_return_code**
The name of a fullword in which the SAF router returns the SAF return code.

**RACF_return_code**
The name of a fullword in which the service routine stores the return code.

**RACF_reason_code**
The name of a fullword in which the service routine stores the reason code.

**Function_code**
The name of a one-byte area containing the function code:

**X'01'** Return RACF user ID

**X'02'** Return DCE UUIDs

**Principal_string_uuid**
The name of an area containing the string form of the principal DCE UUID. The area must be 36 characters long.

- If the function code is **return RACF user ID**, this area must contain the principal DCE UUID as input.
- If the function code is **return DCE UUIDs**, this area is used as an output area when the principal DCE UUID is returned.

**Cell_string_uuid**
The name of an area containing the string form of the cell DCE UUID. The string form of the cell UUID, if supplied by the caller, must be 36 bytes long.

- If the function code is `return RACF user ID`, the *Cell_string_uuid* must be the name of a 36-byte area that contains one of the following:
  - The string form of the cell UUID
  - A null byte (X'00') as the first character of the 36-byte area. If the home cell UUID is not applicable and the caller wants to obtain cross linking information using only the DCE principal UUID, the caller must pass the *Cell_string_uuid* parameter with the first byte of this field containing a null byte of X'00'.
- If the function code is `return DCE UUIDS`, this area is used as an output area when the home cell UUID is returned.

**RACF_userid**
The name of a nine-byte area, which contains a one-byte length field followed by up to 8 characters. It must be specified in uppercase.

When using this callable service to return the RACF user ID associated with a DCE UUID, if the APPLDATA field in the appropriate DCEUUIDS class profile has not been populated with a RACF user ID, the service returns a 0 in the one-byte length field.

## Return and Reason Codes

IRRSUD00 may return the following values in the reason and return code parameters:

| SAF return code | RACF return code | RACF reason code | Explanation |
|---|---|---|---|
| 0 | 0 | 0 | Service successful |
| 4 | 0 | 0 | RACF not installed, or RACF is not active |

| SAF return code | RACF return code | RACF reason code | Explanation |
|---|---|---|---|
| 8 | 8 | 0 | RACF user ID specified does not exist. |
| 8 | 8 | 4 | No mapping to a RACF user ID exists for this UUID. |
| 8 | 8 | 8 | Not authorized to use this service. |
| 8 | 8 | 12 | Internal error during RACF processing |
| 8 | 8 | 16 | Recovery environment could not be established. |
| 8 | 8 | 20 | Local cell DCE UUID could not be determined for this RACF to DCE UUID conversion request. |
| 8 | 8 | 24 | The RACF DCEUUIDS class is not active. UUID to RACF conversion request could not be performed. |
| 8 | 8 | 28 | Parameter list error. |
| 8 | 8 | 32 | No mapping to a UUID exists for this RACF user ID. |

## Usage Notes

1. This callable service allows OS/390 DCE servers to associate a DCE UUID pair with a RACF user ID.
   - If the function code is **return RACF user ID**, the **R_dceruid** service returns the RACF user ID associated with the string form of the DCE UUIDs supplied by the caller, if it is available.
   - If the function code is **return DCE UUID**, the **R_dceruid** service returns the string form of the DCE UUIDs associated with the RACF user ID supplied by the caller, if it is available. RACF stores the UUIDs as uppercase characters and therefore returns the UUIDs in uppercase.

Parameter Usage:

| Parameter | Direction | |
|---|---|---|
| | **UUID to RACF user ID** | **RACF user ID to UUID** |
| SAF_return_code | Output | Output |
| RACF_return_code | Output | Output |
| RACF_reason_code | Output | Output |
| Function_code | Input | Input |
| Principal_string_uuid | Input | Output |
| Cell_string_uuid | Input | Output |
| RACF_userid | Output | Input |

## Related Services

R_dceinfo, R_usermap

# R_exec (IRRSEX00): Set Effective and Saved UIDs/GIDs

## Function

The **R_exec** service sets the effective and saved OS/390 UNIX user identifiers (UIDs) and OS/390 UNIX group identifiers (GIDs) for a process to the specified input values. Input flags indicate whether the UIDs or GIDs or both should be changed.

**R_exec** returns the new values of the real, effective, and saved UIDs and GIDs in the output areas provided.

## Requirements

| | |
|---|---|
| **Authorization:** | Any PSW key in supervisor state |
| **Dispatchable unit mode:** | Task of OS/390 UNIX user |
| **Cross memory mode:** | PASN = HASN |
| **AMODE:** | 31 |
| **RMODE:** | Any |
| **ASC mode:** | Primary or AR mode |
| **Recovery mode:** | ESTAE. Caller cannot have an FRR active. |
| **Serialization:** | Enabled for interrupts |
| **Locks:** | No locks held |
| **Control parameters:** | The parameter list and the work area must be in the primary address space. ALETs must be passed for all parameters except the work area. The words containing the ALETs must be in the primary address space. |

## RACF Authorization

None

## Format

```
CALL IRRSEX00 (Work_area,
               ALET, SAF_return_code,
               ALET, RACF_return_code,
               ALET, RACF_reason_code,
               ALET, Flags,
               ALET, UID,
               ALET, GID,
               ALET, UID_output_area,
               ALET, GID_output_area
              )
```

## Parameters

**Work_area**
The name of a 1024-byte work area for SAF and RACF usage. The work area must be in the primary address space.

**ALET**
The name of a word containing the ALET for the following parameter. Each parameter must have an ALET specified. Each ALET can be different. The words containing the ALETs must be in the primary address space.

**SAF_return_code**
The name of a fullword in which the SAF router returns the SAF return code.

**RACF_return_code**
The name of a fullword in which the service routine stores the return code.

**RACF_reason_code**
The name of a fullword in which the service routine stores the reason code.

**Flags**
The name of a byte containing the settings of the SETUID and SETGID flags for the file being executed. The flags are:

- X'01' - SETUID
- X'02' - SETGID
- X'03' - Both SETUID and SETGID

**UID**
The name of a fullword containing the OS/390 UNIX user identifier (UID) to be set. The UID must be defined to RACF.

**GID**
The name of a fullword containing the OS/390 UNIX group identifier (GID) to be set. The GID must be defined to RACF.

**UID_output_area**
The name of a 3-word area in which the new real, effective, and saved OS/390 UNIX user identifiers (UIDs), in that order, are returned.

**GID_output_area**
The name of a 3-word area in which the new real, effective, and saved OS/390 UNIX group identifiers (GIDs), in that order, are returned.

## Return and Reason Codes

IRRSEX00 returns the following values in the reason and return code parameters:

| SAF Return Code | RACF Return Code | RACF Reason Code | Explanation |
|---|---|---|---|
| 0 | 0 | 0 | The service was successful. |
| 4 | 0 | 0 | RACF is not installed. |
| 8 | 8 | 4 | The OS/390 UNIX user identifier (UID) is not defined to RACF. |
| 8 | 8 | 8 | The OS/390 UNIX group identifier (GID) is not defined to RACF. |
| 8 | 8 | 12 | An internal error occurred during RACF processing. |
| 8 | 8 | 16 | Recovery could not be established. |

## Usage Notes

1. This service is intended only for use by the MVS BCP.
2. An audit record is optionally written, depending on the options in effect for the system.
3. This service uses task level support when OS/390 UNIX has indicated in the task's ACEE that this is a task level process.

### Related Services

None

---

## R_fork (IRRSFK00): Fork a Process

### Function

When called from the parent process, the **R_fork** service returns the address, subpool, and key of the storage containing the user security information for the calling process.

When called from the child process, the **R_fork** service returns the address of an area containing a copy of the security information pointed to on the initial call. The storage pointed to by the address is obtained by the subpool and key returned on the previous call.

### Requirements

| | |
|---|---|
| **Authorization:** | Any PSW key in supervisor state |
| **Dispatchable unit mode:** | Task of OS/390 UNIX user |
| **Cross memory mode:** | PASN = HASN |
| **AMODE:** | 31 |
| **RMODE:** | Any |
| **ASC mode:** | Primary or AR mode |
| **Recovery mode:** | None. Caller handles recovery. |
| **Serialization:** | Enabled for interrupts |
| **Locks:** | No locks held |
| **Control parameters:** | The parameter list and the work area must be in the primary address space. ALETs must be passed for all parameters except the work area. The words containing the ALETs must be in the primary address space. |

### RACF Authorization

None

### Format

```
CALL IRRSFK00   (Work_area,
                 ALET, SAF_return_code,
                 ALET, RACF_return_code,
                 ALET, RACF_reason_code,
                 ALET, Flag,
                 ALET, Data_key,
                 ALET, Data_address,
                 ALET, Data_length
                 ALET, Data_subpool
                )
```

### Parameters

**Work_area**
The name of a 1024-byte work area for SAF and RACF usage. The work area must be in the primary address space.

**ALET**
The name of a word containing the ALET for the following parameter. Each parameter must have an ALET specified. Each ALET can be different. The words containing the ALETs must be in the primary address space.

**SAF_return_code**
The name of a fullword in which the SAF router returns the SAF return code.

**RACF_return_code**
The name of a fullword in which the service routine stores the return code.

**RACF_reason_code**
The name of a fullword in which the service routine stores the reason code.

**Flag**
The name of a fullword flag that describes whether fork processing is being done in the parent's or child's address space:

**X'00000000'**
Fork processing is being done in parent's address space.

**X'00000001'**
Fork processing is being done in child's address space.

**X'00000002'**
Fork processing is being done in parent's address space for additional security information..

**X'00000003'**
Fork processing is being done in child's address space for additional security information..

**Data_key**
The name of a word in which:

- The storage key of the parent's USP or additional security information is returned by IRRSFK00 during parent fork processing.
- The storage key of the parent's USP or additional security information is supplied to IRRSFK00 during child fork processing.

**Data_address**
The name of a word in which the address of:

- The parent's USP or additional security information is returned by IRRSFK00 during parent fork processing.
- A copy of the parent's USP or additional security information is supplied to IRRSFK00 during child fork processing.

**Data_length**
The name of a word that contains the length of the data addressed by Data_address.

**Data_subpool**
The name of a word in which:

- The storage subpool for the parent's USP or additional security information is returned by IRRSFK00 during parent fork processing.
- The storage subpool for the parent's USP or additional security information is supplied to IRRSFK00 during child fork processing.

## Return and Reason Codes

IRRSFK00 returns the following values in the reason and return code parameters:

| SAF Return Code | RACF Return Code | RACF Reason Code | Explanation |
|---|---|---|---|
| 0 | 0 | 0 | The service was successful. |
| 0 | 0 | 4 | Additional security information available. |

## Usage Notes

1. This service is intended only for use by the MVS BCP.
2. The key is meaningful only when the following subpools are used:
   - 129–132
   - 227–231
   - 241
3. The following user security information is propagated from the parent address space to the child address space:
   - The user security packet (USP)
   - The privilege bit (ACEEPRIV) from the parent's ACEE
   - The privilege bit (TOKPRIV) and the trusted bit (TOKTRST) from the parent's TOKEN
   - Additional security information. For RACF, this includes:
     - Controlled status
     - Keep-controlled indicators
     - Saved messages
   - RACF reason code 4 indicates that there is additional security information related to the parent address space that should be propagated to the child address space. If a reason code 4 is received when a flag value of 0 was passed, R_fork should be called again with a flag value of 2 specified. If a reason code 4 is received when a flag value of 1 was passed, R_fork should be called again with a flag value of 3 specified.
4. This service uses task level support when OS/390 UNIX has indicated in the task's ACEE that this is a task level process.

## Related Services

None

# R_getgroups (IRRSGG00): Get/Set Supplemental Groups

## Function

The **R_getgroups** service checks the high-order bit of the input group count. See **Group_count** under "Parameters" on page 135 for more information.

The GIDs are not explicitly added to or deleted from the supplemental group list. A GID is in this list if the user was a member of the group when the user's ACEE was created through a RACROUTE REQUEST=VERIFY request and if the GID was assigned to the group before the **initUSP** service was performed for the process.

## Requirements

| | |
|---|---|
| **Authorization:** | Any PSW key in supervisor state |
| **Dispatchable unit mode:** | Task of OS/390 UNIX user |
| **Cross memory mode:** | PASN = HASN |
| **AMODE:** | 31 |
| **RMODE:** | Any |
| **ASC mode:** | Primary or AR mode |
| **Recovery mode:** | SETFRR |
| **Serialization:** | Enabled for interrupts |
| **Locks:** | No locks held |
| **Control parameters:** | The parameter list and the work area must be in the primary address space. ALETs must be passed for all parameters except the work area. The words containing the ALETs must be in the primary address space. |

## RACF Authorization

None

## Format

```
CALL IRRSGG00 (Work_area,
               ALET, SAF_return_code,
               ALET, RACF_return_code,
               ALET, RACF_reason_code,
               ALET, User_key,
               ALET, Group_count,
               ALET, Grouplist,
               ALET, Number_of_GIDs
              )
```

## Parameters

**Work_area**
The name of a 1024-byte work area for SAF and RACF usage. The work area must be in the primary address space.

**ALET**
The name of a word containing the ALET for the following parameter. Each parameter must have an ALET specified. Each ALET can be different. The words containing the ALETs must be in the primary address space.

**SAF_return_code**
The name of a fullword in which the SAF router returns the SAF return code.

**RACF_return_code**
The name of a fullword in which the service routine stores the return code.

**RACF_reason_code**
The name of a fullword in which the service routine stores the reason code.

**User_key**
The name of a byte containing the user's key. This key is used to store into the output grouplist area. The key is in the four high-order bits of the byte.

**Group_count**
The name of a word containing the number of GID entries that can be stored in the *Grouplist* area. IRRSGG00 uses the high-order bit to determine how to process the value in the parameters.

If the high-order bit of the input *Group_count* is:

1. On, the caller must store into this area the list of GIDs of the supplemental groups to be set as the supplemental groups of the current process.
2. Off, IRRSGG00 checks the input *Group_count* value. If it is:
   a. 0, the *Grouplist* area is not used. IRRSGG00 returns the total supplemental GID count of the current process in the *Number_of_GIDs* parameter.
   b. Less than the total supplemental GID count:
      1) An error code is returned.
      2) The GIDs of the supplemental groups for the current process are put into the *Grouplist* area, which can only accommodate the number of GIDs specified in the *Group_count* parameter.
      3) The count of the supplemental GIDs actually placed in the *Grouplist* area is returned in the *Number_of_GIDs* parameter.
      4) The *Group_count* field is set to the total supplemental GID count of the current process.

         The supplemental groups in the *Grouplist* area are listed in the same order as the group connections shown in the output of the LISTUSER command.
   c. Greater than or equal to the total supplemental GID count:
      1) The GIDs of the supplemental groups for the current process are put into the *Grouplist* area.
      2) The supplemental GID count of the current process is put into the *Number_of_GIDs* parameter.

**Grouplist**
The name of an area in which the GIDs of the supplemental groups for a process are returned. The *Group_count* parameter indicates the number of entries this area can contain. The GIDs are returned as consecutive 4-byte entries.

**Number_of_GIDs**
The name of a word in which the number of GIDs put in the *Grouplist* area is returned.

## Return and Reason Codes

IRRSGE00 returns the following values in the reason and return code parameters:

| SAF Return Code | RACF Return Code | RACF Reason Code | Explanation |
|---|---|---|---|
| 0 | 0 | 0 | The service was successful. |
| 4 | 0 | 0 | RACF is not installed. |
| 8 | 8 | 4 | *Group_count* is less than the number of supplemental groups (see item 2b under the **Group_count** parameter). |
| 8 | 8 | 8 | The grouplist address is not valid. |

| SAF Return Code | RACF Return Code | RACF Reason Code | Explanation |
|---|---|---|---|
| 8 | 8 | 12 | An internal error occurred during RACF processing. |

## Usage Note

- This service is intended only for use by the MVS BCP and by OS/390 UNIX System Services servers. The service contains support for OS/390 UNIX System Services servers, but cannot be directly invoked by an OS/390 UNIX System Services server.

## Related Services

None

---

# R_getgroupsbyname (IRRSUG00): Get Groups by Name

## Function

The **R_getgroupsbyname** service checks the input group count and, if it is zero, returns the number of supplemental groups for the specified user ID. If the input count is not zero and it is less than the number of groups, an error code is returned. If the count is not less than the number of groups, the GIDs of the supplemental groups for the specified user ID are put into the grouplist area, and the number of GIDs is returned.

## Requirements

**Authorization:** Any PSW key in supervisor state
**Dispatchable unit mode:** Task of OS/390 UNIX user
**Cross memory mode:** PASN = HASN
**AMODE:** 31
**RMODE:** Any
**ASC mode:** Primary or AR mode
**Recovery mode:** ESTAE. Caller cannot have an FRR active.
**Serialization:** Enabled for interrupts
**Locks:** No locks held
**Control parameters:** The parameter list and the work area must be in the primary address space. ALETs must be passed for all parameters except the work area. The words containing the ALETs must be in the primary address space.

## RACF Authorization

1. A RACF user can be connected to more than NGROUPS_MAX groups, but only up to the first NGROUPS_MAX OS/390 UNIX groups will be associated with the user for this service.

   The first NGROUPS_MAX OS/390 UNIX groups to which a user is connected, as shown by a LISTUSER command, are the groups that get associated with the user.

## Format

```
CALL IRRSUG00 (Work_area,
               ALET, SAF_return_code,
               ALET, RACF_return_code,
               ALET, RACF_reason_code,
               ALET, User_key,
               ALET, Userid_len,
               ALET, Userid,
               ALET, Group_count,
               ALET, Grouplist,
               ALET, Number_of_GIDs
              )
```

## Parameters

**Work_area**
The name of a 1024-byte work area for SAF and RACF usage. The work area must be in the primary address space.

**ALET**
The name of a word containing the ALET for the following parameter. Each parameter must have an ALET specified. Each ALET can be different. The words containing the ALETs must be in the primary address space.

**SAF_return_code**
The name of a fullword in which the SAF router returns the SAF return code.

**RACF_return_code**
The name of a fullword in which the service routine stores the return code.

**RACF_reason_code**
The name of a fullword in which the service routine stores the reason code.

**User_key**
The name of a byte containing the user's key. This key is used to store into the output grouplist area and number_of GIDs word. The key is in the four high-order bits of the byte.

**Userid_length**
The name of a byte containing the length of the user ID.

**Userid**
The name of an 8-byte area containing the user ID whose groups are to be returned. The user ID must be left-justified in the area. The userid_length parameter specifies the actual length of the name.

**Group_count**
The name of a fullword containing the number of GID entries in the input grouplist area.

**Grouplist**
The name of an area in which the GIDs of the supplemental groups are returned. The GIDs are returned as consecutive 4-byte entries. The group_count parameter indicates the number of entries this area can contain.

**Number_of_GIDs**
The name of a word in which the number of GIDs actually put in the grouplist area is returned.

## Return and Reason Codes

IRRSUG00 returns the following values in the reason and return code parameters:

| SAF Return Code | RACF Return Code | RACF Reason Code | Explanation |
|---|---|---|---|
| 0 | 0 | 0 | The service was successful. |
| 4 | 0 | 0 | RACF is not installed. |
| 8 | 8 | 4 | The group count is less than number of supplemental groups. |
| 8 | 8 | 8 | The grouplist address is not valid. |
| 8 | 8 | 12 | An internal error occurred during RACF processing. |
| 8 | 8 | 16 | Recovery could not be established. |
| 8 | 8 | 20 | RACROUTE VERIFY processing failed. |
| 8 | 8 | 24 | The user ID is not defined to RACF. |

## Usage Notes

1. This service is intended only for use by the MVS BCP.

## Related Services

None

# R_IPC_ctl (IRRSCI00): Perform IPC Control

## Function

The **R_IPC_ctl** service performs a function based on the function code value in the parameter list.

- When the function is Check Owner for Remove ID, **R_IPC_ctl** checks whether the current process has the appropriate authority to the IISP.
- When the function is Check Owner and Set, **R_IPC_ctl** sets the owner's UID and GID and the mode permission bits if the current process has the appropriate authority.
- When the function is Check Superuser and Set, **R_IPC_ctl** sets the owner's UID and GID and the mode permission bits if the current process is a superuser.

## Requirements

**Authorization:** Any PSW key in supervisor state
**Dispatchable unit mode:** Task of OS/390 UNIX user/any task if system user type is specified
**Cross memory mode:** PASN = HASN or PASN not = HASN
**AMODE:** 31
**RMODE:** Any
**ASC mode:** Primary or AR mode
**Recovery mode:** SETFRR
**Serialization:** Enabled for interrupts
**Locks:** No locks held

**Control parameters:**

The parameter list and the work area must be in the primary address space. ALETs must be passed for all parameters except the work area. The words containing the ALETs must be in the primary address space.

## RACF Authorization

1. The access checks performed are defined in XPG4 System Interfaces and Headers under msgctl, semctl, and shmctl interfaces for commands IPC_SET and the IPC_RMID. The access checks are as follows:
   a. The effective OS/390 UNIX user identifier (UID) for the calling process is used for all access checks.
   b. If the CREI user type is system, IRRSCI00 grants authorization when the function is Check Owner for Remove ID, or updates the IPCP when the function is either Check Owner and Set or Check Superuser and Set.
   c. The user is considered a superuser if the effective UID is zero or if the ACEE indicates trusted or privileged authority.
   d. If the function is Check Owner for Remove ID, the user must be either a superuser or the effective UID of the process must match either the owner's UID or creator's UID in the IISP for a successful completion. Otherwise, the user is not authorized.

      **Note:** If the caller is unauthorized as stated above, an authorization check is performed on the resource name in the UNIXPRIV class indicated in Table 71. If the authorization check is successful, the caller is treated as a superuser.

*Table 71. UNIXPRIV class resource names used in R_IPC_ctl*

| Function code | Resource name | Access required |
|---|---|---|
| 1-Check Owner for Remove ID | SUPERUSER.IPC.RMID | READ |

2. If the function is Check Superuser and Set, the user must be a superuser in order to set the owner's OS/390 UNIX user identifier (UID), owner's OS/390 UNIX group identifier (GID), and mode fields from the input parameters into the IISP for a successful completion. Otherwise, the user is not authorized.
3. If the function is Check Owner and Set, the user must be either a superuser or the effective UID of the process must match either the owner's UID or creator's UID in the IISP in order to set the owner's UID, owner's GID, and mode fields from the input parameters into the IISP for a successful completion. Otherwise, the user is not authorized.

## Format

```
CALL IRRSCI00 (Work_area,
               ALET, SAF_return_code,
               ALET, RACF_return_code,
               ALET, RACF_reason_code,
               ALET, Function_code,
               ALET, Owner_UID,
               ALET, Owner_GID,
               ALET, Mode_Permissions,
               ALET, ISP,
               ALET, CREDIPC
              )
```

## Parameters

**Work_area**
The name of a 1024-byte work area for SAF and RACF usage. The work area must be in the primary address space.

**ALET**
The name of a word containing the ALET for the following parameter. Each parameter must have an ALET specified. Each ALET can be different. The words containing the ALETs must be in the primary address space.

**SAF_return_code**
The name of a fullword in which the SAF router returns the SAF return code.

**RACF_return_code**
The name of a fullword in which the service routine stores the return code.

**RACF_reason_code**
The name of a fullword in which the service routine stores the reason code.

**Function_code**
The name of a 1-byte area containing the function code:

**X'01'** Check Owner for Remove ID

**X'02'** Check Owner and Set

**X'03'** Check Superuser and Set

**Owner_UID**
The name of a 4-byte area containing the new owner's UID to be set.

**Owner_GID**
The name of a 4-byte area containing the new owner's GID to be set.

**Mode_Permissions**
The name of a 4-byte area containing the new mode permission bits to be set. The following is a list of defined permission bits mapped by BPXYMODE:

**S_IRUSR**
Permits the process that owns the IPC member to read it.

**S_IWUSR**
Permits the process that owns the IPC member to alter it.

**S_IRGRP**
Permits the group associated with the IPC member to read it.

**S_IWGRP**
Permits the group associated with the IPC member to alter it.

**S_IROTH**
Permits others to read the IPC member.

**S_IWOTH**
Permits others to alter the IPC member.

Alter and write have the same meaning for access checks. Alter applies to semaphores, and write applies to message queueing and shared memory segments.

**ISP**
The name of the IISP for the file being accessed.

**CREDIPC**
The name of the CREI structure for the current IPC system callable service. The CREI contains the IPC identifier and the IPC key. See *OS/390 SecureWay Security Server RACF Data Areas*.

## Return and Reason Codes

IRRSCI00 may return the following values in the reason and return code parameters:

| SAF return code | RACF return code | RACF reason code | Explanation |
|---|---|---|---|
| 0 | 0 | 0 | The service was successful. |
| 4 | 0 | 0 | RACF is not installed. |
| 8 | 8 | 4 | The user is not authorized. |
| 8 | 8 | 12 | An internal error occurred during RACF processing. |
| 8 | 8 | 32 | The CREI user type is not supported. |

## Usage Notes

1. This service is intended for use only by the MVS BCP.
2. An audit record is optionally written, depending on the audit options in effect for the system.
3. This service uses task level support when OS/390 UNIX has indicated in the task's ACEE that this is a task level process.

## Related Services

makeISP, ck_IPC_access

---

# R_kerbinfo (IRRSMK00): Retrieve or Set SecureWay Security Server Network Authentication and Privacy Service Fields

## Function

The **R_kerbinfo** callable service can be used to either retrieve RACF Network Authentication and Privacy Service information, or to update the count of unsuccessful attempts to use a Network Authentication and Privacy Service key.

The action performed by this callable service is based on the function code passed by the caller in the R_kerbinfo parameter list:

- When the function code is set to X'01', R_kerbinfo retrieves local Network Authentication and Privacy Service principal information. The caller must identify the principal by providing a RACF name or a Network Authentication and Privacy Service principal name.
- When the function code is set to X'02', R_kerbinfo increments the count of invalid attempts by a Network Authentication and Privacy Service principal to use a key. The caller must identify the principal by providing a Network Authentication and Privacy Service principal name.
- When the function code is set to X'03', R_kerbinfo resets the count of invalid attempts by a Network Authentication and Privacy Service principal to use a key to zero. The caller must identify the principal by providing a Network Authentication and Privacy Service principal name.
- When the function code is set to X'04', R_kerbinfo retrieves Network Authentication and Privacy Service realm information. The caller may identify the realm by providing a Network Authentication and Privacy Service realm profile name, or by providing a NULL name, in which case RACF will return information about the local realm, KERBDFLT.

Field data is returned to the invoker in a data structure containing a set of repeating ordered triplets, which are of the format name of the field, length of the field, and field data.

## Requirements

| | |
|---|---|
| **Authorization:** | Any PSW key in supervisor state |
| **Dispatchable unit mode:** | Any task |
| **Cross memory mode:** | PASN = HASN |
| **AMODE:** | 31 |
| **RMODE:** | Any |
| **ASC mode:** | Primary only |
| **Recovery mode:** | ESTAE. Caller cannot have an FRR active. |
| **Serialization:** | Enabled for interrupts |
| **Locks:** | No locks held |
| **Control parameters:** | The parameter list and the work area must be in the primary address space. The words containing the ALETs must be in the primary address space. |

## Linkage Conventions

The parameter list for this callable service is intended to be variable length to allow for future expansion. To allow for this, the last word in the parameter list must have a one in the high-order (sign) bit.

## Format

```
CALL IRRSMK00 (Work_area,
               ALET, SAF_return_code,
               ALET, RACF_return_code,
               ALET, RACF_reason_code,
               Function_code,
               RACF_name,
               KERB_name,
               Data_area,
              )
```

## Parameters

**Work_area**
The name of a 1024-byte work area for SAF and RACF usage. The work area must be in the primary address space.

**ALET**
The name of a word containing the ALET for the following parameter. Each parameter must have an ALET specified. Each ALET can be different. The words containing the ALETs must be in the primary address space.

**SAF_return_code**
The name of a full word in which the SAF router returns the SAF return code.

**RACF_return_code**
The name of a full word in which the service routine stores the return code.

**RACF_reason_code**
The name of a full word in which the service routine stores the reason code.

**Function_code**
The name of a one-byte area in the primary address space containing the function code:

**X'01'**
Retrieve local Network Authentication and Privacy Service principal information.

**X'02'**
Increment a Network Authentication and Privacy Service principal's count of invalid key attempts.

RACF will process this request much the same as it does when an invalid password is supplied during TSO logon. When the number of attempts exceeds the number of incorrect password attempts which RACF allows, set using the SETROPTS command PASSWORD REVOKE suboperand, the RACF user ID will be revoked, an ICH408I message will be issued and an SMF Type 80 record will be written.

**X'03'**
Reset a Network Authentication and Privacy Service principal's count of invalid key attempts to zero.

RACF will process this request much the same as a successful TSO logon request and will update the date/time of the last successful logon (the LJDATE/LJTIME fields in the RACF user profile) to the current date/time.

**X'04'**
Retrieve Network Authentication and Privacy Service realm information.

**RACF_name**
The name of a 9-byte area in the primary address space consisting of a one-byte length field followed by up to 8 characters. If a value is specified for RACF_name, it must be defined RACF user ID and specified in uppercase. If a RACF user ID is not specified, the length must equal zero.

RACF_name may only be specified with function code X'01' and will be used to identify a local Network Authentication and Privacy Service principal by the principal's RACF user identity.

**KERB_name**
The name of an area in the primary address space containing the 240 byte

Network Authentication and Privacy Service name. The name must be left-justified and padded with blanks. The only way to get the local realm information is to specify a null in the KERB_name field. If a Network Authentication and Privacy Service realm profile name is specified, it must be realm qualified (for example, follow the DCE-like convention of /.../REALM_A/KRBTGT/REALM_B) and must be folded to all uppercase.

**Note:** Local Network Authentication and Privacy Service principal names and realm names returned in the Data_area NAME field will not be realm qualified and will be case sensitive.

If the caller does not wish to specify a KERB_name, then the first character of the area must be a NULL byte (that is, the first byte of the 240 byte area must contain X'00').

KERB_name may be specified with any function code. For function codes X'01,',X'02', and X'03', it is used to identify a local Network Authentication and Privacy Service principal. For function X'04', it is used to identify a Network Authentication and Privacy Service realm profile name.

**Data_area**

The name of an area in the primary address space for fields to be retrieved. The format of the Data_area structure is:

| *Offset* | *Length* | *Description* |
|---|---|---|
| 0 | 2 | The length in bytes of the entire Data_area structure |
| 2 | 2 | Total number of fields |
| 4 | 8 | The name of the field |
| 12 | 2 | The length of the field data |
| 14 | variable | Field data |

The ordered triplet (name of the field, length of the field, and field data)is a repeating data structure. This triplet will repeat for the total number of fields in the input Data_area.

The following table lists the fields that will be returned for function code X'01' (retrieve local principal information) and X'04' (retrieve realm information). The caller supplied Data_area structure must be allocated large enough for all of the fields associated with the function to be returned. The fields that will be returned, along with each field's maximum length in bytes and the order that they will be returned, can be found in the following table:

| **Field Name** | **Maximum Length** | **Data Type** | **Description** |
|---|---|---|---|
| USERID | 8 | Character | RACF userid (N/A for function X'04') |
| REVOKED | 1 | Boolean | Flag indicating user has been revoked (N/A for function X'04') |
| EXPIRED | 1 | Boolean | Flag indicating user's password has expired (N/A for function X'04') |
| NAME | 240 | Character | Kerberos name |
| MINTKTLF | 4 | Integer | Minimum ticket life value (N/A for function X'01') |

| Field Name | Maximum Length | Data Type | Description |
|---|---|---|---|
| MAXTKTLF | 4 | Integer | Maximum ticket life value |
| DEFTKTLF | 4 | Integer | Default ticket life value (N/A for function X'01') |
| SALT | 240 | Character | Current key salt |
| ENCTYPE | 4 | Binary | Will have a value of X'00000001' if key has been defined. |
| CURKEYV | 1 | Integer | Current key version (1-255) |
| CURKEY | 10 | Character | Current key - returned as 2 byte length, followed by key value (8 bytes) |
| PREVKEYV | 1 | Integer | Previous key version (1-255) |
| PREVKEY | 10 | Character | Previous key - returned as 2 byte length, followed by key value (8 bytes) |

The minimum length of the Data_area structure is then 662 bytes, accounting for the two 2-byte length fields at the beginning and 13 sets of field triplets, each with the above length requirements.

## Return and Reason Codes

IRRSMK00 may return the following values in the reason and return code parameters:

| SAF Return Code | RACF Return Code | RACF Reason Code | Explanation |
|---|---|---|---|
| 0 | 0 | 0 | The service was successful. |
| 4 | 0 | 0 | RACF is not installed. |
| 4 | 4 | 0 | User revoked prior to call. |
| 4 | 4 | 4 | User revoked by this call. |
| 8 | 8 | 0 | Invalid function code. |
| 8 | 8 | 4 | Parameter list error. |
| 8 | 8 | 8 | Internal error during RACF processing. |
| 8 | 8 | 12 | Recovery environment could not be established. |
| 8 | 8 | 20 | RACF profile does not have required segments. |
| 8 | 8 | 24 | Length of the output area is too small to contain the data retrieved. |
| 8 | 8 | 32 | RACF profile specified does not exist. |
| 8 | 8 | 36 | Mapping to RACF profile failed. |

## Usage Notes

1. The caller is in supervisor state, so the task and address space ACEEs are not checked. Therefore, for example, an authorized caller may extract KERB segment fields, or update the invalid key count, for any user who has a KERB segment.

2. This service returns fields that have been previously populated. Associated with the returned fields is a length indicator. The length indicator is set to zero if a field does not exist.
3. If RACF_name and KERB_name are both provided for function X'01', R_kerbinfo will use RACF_name.
4. If RACF_name is provided for any function other than X'01', a parameter list error will be returned.
5. If KERB_name is not supplied on a function X'04' request (the first character is NULL), information about the local OS/390 SecureWay Kerberos Security Server, KERBDFLT, will be returned. Alternatively, KERB_name may be explicitly set to KERBDFLT.
6. It is the responsibility of the caller to obtain and free the Data_area. If the fields to be retrived from RACF are larger than the Data_area, RACF fails the request.
7. Field level access checking does not occur when retrieving fields with this service.
8. Field names are returned as 8-character fields, left-justified, and padded with blanks. They are specified in uppercase.
9. Fields that are not applicable for a function code, such as USERID for function code X'04', will be returned with the length set to zero.
10. If function code X'02' causes a user to be revoked, an ICH408I message will be issued and an SMF Type 80 record will be cut.

## Parameter Usage

*Table 72. Parameter Usage*

| Parameter | Function X'01' | Function X'02' | Function X'03' | Function X'04' |
|---|---|---|---|---|
| SAF_return_code | Output | Output | Output | Output |
| RACF_return_code | Output | Output | Output | Output |
| RACF_reason_code | Output | Output | Output | Output |
| Function_code | Input | Input | Input | Input |
| KERB_name | Input | Input | Input | Input |
| Data_area | Input/Output | N/A | N/A | Input/Output |

## Related Services

R_ticketserv, R_usermap

# R_PKIServ(IRRSPX00): Request Public Key Infrastructure (PKI) Services

## Function

The R_PKIServ SAF callable service allows applications to request the generation and retrieval of X.509 V.3 certificates.

## Requirements

**Authorization:** Any PSW key in supervisor or problem state

| | |
|---|---|
| **Dispatchable unit mode:** | Task of user |
| **Cross memory mode:** | PASN = HASN |
| **AMODE:** | 31 |
| **RMODE:** | Any |
| **ASC mode:** | Primary or AR mode |
| **Recovery mode:** | ESTAE. Caller cannot have a FRR active. |
| **Serialization:** | Enabled for interrupts |
| **Locks:** | No locks held |
| **Control parameters:** | The parameter list and the work area must be in the primary address space. The words containing the ALETs must be in the primary address space. |

## RACF Authorization

This interface is protected by FACILITY class profiles of the form IRR.RPKISERV.*(function)*, where *(function)* is one of the names descripted under **Function_code** below. The user ID (from the ACEE associated with the address space) for the application, is used to determine access:

**NONE**
Access is denied.

**READ**
Access is permitted based on subsequent access checks against the caller's user ID. To determine the caller, the current TCB is checked for an ACEE. If one is found, the authority of that user is checked. If there is no ACEE associated with the current TCB, the ACEE associated with the address space is used to locate the user ID.

**UPDATE**
Access is permitted based on subsequent access checks against the application's user ID.

**ALTER OR CONTROL (or user ID is RACF SPECIAL)**
Access is permitted with no subsequent access checks made.

For READ and UPDATE access, subsequent access checks are performed against the IRR.DIGTCERT.(function) FACILITY profiles. These are identical to the checks made by the RACDCERT TSO command. See the *OS/390 SecureWay Security Server RACF Command Language Reference, SC28-1919-07* and the *OS/390 SecureWay Security Server RACF Security Administrator's Guide, SC28-1915-07* for more information.

## Format

```
CALL IRRSPX00 (Work_area,
               ALET, SAF_return_code,
               ALET, RACF_return_code,
               ALET, RACF_reason_code,
                         Number_parameters,
                         Function_code,
                         Attributes,
                         Log_string,
                         Parm_list_version,
                         Function_parmlist
                )
```

## Parameters

**Work_area**
The name of a 1024-byte work area for SAF. The work area must be in the primary address space.

**ALET**
The name of a word containing the ALET for the following parameter. Each parameter must have an ALET specified. Each ALET can be different. The words containing the ALETs must be in the primary address space.

**SAF_Return_Code**
The name of a fullword in which the SAF router returns the SAF return code.

**RACF_Return_Code**
The name of a fullword in which the service routine stores the return code.

**RACF_Reason_Code**
The name of a fullword in which the service routine stores the reason code.

**Number_parameters**
Specifies the name of a fullword which contains the number of parameters that follow in the non-request specific portion of the R_PKIServ callable service invocation.

**Function_code**
The name of a 2-byte area containing the Function code. The function code has one of the following values:

- X'0001'-Generate a basic X.509V3 certificate using the application provided data pointed to by the function specific parameter list (Function name GENCERT).
- X'0002'-Extract certificate by certificate request ID (Function name EXPORT).

**Attributes**
The name of a 4-byte area which must contain binary zeros. This is a reserved field that must be specified.

**Log_string**
The name of an area that consists of a 1-byte length field followed by character data to be included in any audit records that are created as a result of the R_PKIServ invocation. This includes the IRRSPX00-specific audit event codes, as well as any audit records created as a result of checking profiles in the FACILITY class. If not specified, the length must equal 0.

**Parmlist_version**
The name of a 4-byte input value which contains the version number for the following input field, Function_parmlist. The contents of this field must be set to binary zero.

**Function_parmlist**
Specifies the name of the function code specific parameter list for the Function_code specified:

*Table 73. GENCERT*

| Field | Attributes | Usage | Description |
|---|---|---|---|
| Eyecatcher | 8 characters | In | Eyecatcher, 8 characters left justified blank filled. Actual value set by invoker, for example 'GENCERT'. |

*Table 73. GENCERT (continued)*

| Field | Attributes | Usage | Description |
|---|---|---|---|
| CertPlistLen | 4-byte length | In | Describes the length in bytes of the certificate generation plist. |
| CertPlist | Address of | In | The name of the area which is the CertGen request parameter list. This area maps the specific name, length, address/data values which are used in satisfying the certificate request for the specified user. |
| Certid | Address of | In/Out | Points to a 57-byte area, in which the first byte will contain the actual length on return of the certificate request ID. The storage address specified must be obtained by the caller, and freed by the caller. The returned certificate request ID is used to extract the completed certificate, if the request has been accepted by RACF. |

The GENCERT function has in essence two connected parameter list areas; the function specific parameter list as defined above and the CertGen request parameter list (CertPlist) containing specific certificate field information. CertPlist is a list of ordered triplets which consists of name, length, and data value. The field name is a fixed 12 character field, case sensitive, left justified, and padded with blanks, the length field is a binary four byte value, which qualifies the length of the data item. Note that all data values are EBCDIC character data unless otherwise indicated. The following tables describes the valid certificate request fields:

| Field Name | Max Length | Description |
|---|---|---|
| DiagInfo | 80 bytes (exactly) | EyeCatcher to identify this request in virtual storage for diagnostic reasons. For certificate generation warnings and errors, RACF will update the field with diagnostic information. The length will also be updated. Required Field must be the first field in the CertPlist. |
| CommonName | 64 bytes | Subject's common name. This is optional. No default, except if specified with a null value (length 0), RACF will use the PGMRNAME field from the RACF user profile as determined by the UserID field for this request. If PGMRNAME is null, the common name will be of the form of RACF UserID:*(user's-racf-identify)*, for example RACF UserID:JDOE |
| Title | 64 bytes | Subject's Title. This is optional. No default. |
| OrgUnit | 64 bytes | Subject's Organizational Unit. Note that this field may be repeated. RACF concatenates in the order of appearance to construct the hierarchy of organizational units. This is optional. No default. |
| Org | 64 bytes | Subject's Organization. This is optional. No default. |
| Locality | 64 bytes | Subject's City or Locality. This is optional. No default. |
| StateProv | 64 bytes | Subject's State/Providence. This is optional. No default. |

| Field Name | Max Length | Description |
|---|---|---|
| Country | 64 bytes | Subject's Country. This is optional. No default. |
| KeyUsage | 20 bytes | One of 'handshake'\|'dataencrypt'\|'certsign'\|'docsign' (case insensitive, do not use quotes). Note this field may be repeated to request multiple usages. This is optional. No default. |
| NotBefore | 2 bytes | Number of days from today's date that the certificate becomes valid. Range is 0-30. Validity checked by RACF. This is optional. Default is 0. |
| NotAfter | 4 bytes | Number of days from today's date that the certificate expires. Range is 90-3650. Validity checked by RACF. This is optional. Default is 365. |
| AltIPAddr | 15 bytes | Dotted decimal V4 IP address. This is optional. No default. |
| AltURI | 255 bytes | Uniform Resource Identifier. This is optional. No default. |
| AltEmail | 100 bytes | E-mail address. This is optional. No default. |
| AltDomain | 100 bytes | Domain Name. This is optional. No default. |
| UserId | 8 bytes | Subject's RACF UserID. If not specified, the userID is taken from the ACEE. |
| Label | 32 bytes | Up to 32 mixed case characters which may be used as the 'handle'. This is optional. Default is one will be generated and added to the user's list of certificates. |
| SignWith | 45 bytes | Label of OS/390 Certificate authority certificate to sign the completed certificate request. The format is SAF:CERTAUTH/(ca-cert-label) or SAF:/(ca-cert-label), where ca-cert-label is the certificate label under CERTAUTH or the caller's UserID. This is a required field. |
| PublicKey | 65535 bytes | PKCS #10 or Netscape Navigator certificate request containing the public key to be certified. This is base64 encoded DER. |

*Table 74. EXPORT*

| Field | Attributes | Usage | Description |
|---|---|---|---|
| Eyecatcher | 8 characters | In | Eyecatcher, 8 characters left justified blank filled. Actual value set by invoker, for example 'EXPORT'. |
| CertAnchorLen | 4 byte length | In/Out | 4-byte area which is the length of the preallocated storage of the CertAnchor area on input to EXPORT. RACF will update this value with the actual length of the certificate returned. In the event that the storage area as specified by the CertAnchor address is too small, RACF will set a failing return/reason code and update the length field to the size required. The caller must allocate a larger area. |

*Table 74. EXPORT (continued)*

| Field | Attributes | Usage | Description |
|---|---|---|---|
| CertAnchor | Address of | In/Out | The address of the storage area in which the R_PKIServ service stores the certificate that is specified by the CertID parameter if the service was able to successfully retrieve the completed certificate. If the caller has supplied an area which is too small, based on the CertAnchorLen, this service fails the request, and updates the CertAnchorLen field to indicate the actual storage required to store the certificate. |
| CertId | Address of | In | Points to a 57-byte area, in which the first byte will contain the actual length of the input certificate request ID that will be used to locate the certificate to be exported. |

## Return and Reason Codes

R_PKIServ may return the following values in the reason and return code parameters:

*Table 75. Return and Reason Codes*

| SAF Return Code | RACF Return Code | RACF Reason Code | Explanation |
|---|---|---|---|
| 0 | 0 | 0 | Successful completion |
| 4 | 0 | 0 | RACF not installed |
| 8 | 8 | 4 | A parameter list error has been detected. See Usage Notes for further details |
| 8 | 8 | 8 | The caller of this service has not been RACF authorized to use this callable service |
| 8 | 8 | 12 | An internal error has occurred during RACF processing of the requested function |
| 8 | 8 | 16 | Unable to establish a recovery environment |
| 8 | 8 | 20 | Function code specified is not defined |
| 8 | 8 | 24 | Parameter list version specified is not supported |

Reason and return code parameters specific to function GENCERT:

| SAF Return Code | RACF Return Code | RACF Reason Code | Explanation |
|---|---|---|---|
| 8 | 8 | 40 | CertPlist has an incorrect length |
| 8 | 8 | 44 | CertPlist DiagInfo field missing or has an incorrect length |
| 8 | 8 | 48 | Incorrect field name specified in CertPlist |
| 8 | 8 | 52 | Incorrect field value specified in CertPlist |

| SAF Return Code | RACF Return Code | RACF Reason Code | Explanation |
|---|---|---|---|
| 8 | 8 | 56 | Required field missing from CertPlist |
| 8 | 8 | 60 | Certificate generation provider input or environment error |

Reason and return code parameters specific to function EXPORT:

| SAF Return Code | RACF Return Code | RACF Reason Code | Explanation |
|---|---|---|---|
| 8 | 8 | 40 | CertAnchor area missing |
| 8 | 8 | 44 | CertAnchor area too small |
| 8 | 8 | 48 | Incorrect CertID specified |

## Usage Notes

1. This service is intended for use by OS/390 application servers, to request the fulfillment of a X.509 V.3 certificate request.
2. An audit record will be created as a result of a certificate generation request or retrieval which will indicate the success or failure of the attempt.
3. The input parameter list for GENCERT function consists of triplets which consist of field name, field length, and data. The field name is a fixed field, 12 characters in length, and the field name must be left justified, and padded with blanks. The data length is also a fixed width field of 4 bytes which contain an integer which represents the length of the following field data.
4. The R_PKIServ service requires the caller to preallocate the 57 byte storage area which will hold the certificate ID which will be returned on a successful GENCERT. On a successful GENCERT, RACF will update the first byte with the actual length of the CertID. The entire 57 area must be provided for EXPORT even if the actual CertID is smaller than that.
5. The R_PKIServ service requires the caller to preallocate the storage which will hold the certificate which is being extracted via the EXPORT function code. On successful certificate retrieval, RACF will update the CertAnchorLen field with the actual length of the certificate. If the storage area is too small to hold the certificate, RACF will fail the request and update the CertAnchorLen field in the EXPORT request specific parameter list as supplied by the caller of this service. The caller is responsible for releasing and obtaining a new area of virtual storage which is the size as specified by RACF, and retrying the EXPORT operation.
6. The actual values for the eyecatchers in the function specific parameters lists and the DiagInfo field in the CertPlist for GENCERT invocations are determined by the caller.
7. For GENCERT CertPlist field errors (reason codes 48, 52, and 56), RACF will update the DiagInfo field with the name of the field in error. The length will also be updated.
8. For GENCERT certificate generation provider errors (reason code 60), RACF will update the DiagInfo field with product specific diagnostic data. This data has the following format: **error-description (message-ID)**, where message-ID is the RACDCERT error message ID that is closely related to the following error:

```
No matching certificate was found for "SignWith" (IRRD107I)
"PublicKey" encoding does not have a valid signature (IRRD112I)
"PublicKey" encoding is not valid (IRRD104I)
"PublicKey" encoding contains an unsupport encryption algorithm (IRRD118I)
"PublicKey" extension not permitted for CERTAUTH certificate (IRRD126I)
"Label" specified is already in use (IRRD111I)
"SignWith" requires a certificate with an associated private key (IRRD128I)
Certificate cannot be added.  Serial number for this CA already in use (IRRD109I)
SignWith key is an ICSF key.  ICSF is not operational (IRRD135I)
Subject's name exceeds the maximum allowed characters, which is 255 (IRRD131I)
```

Additionally, for successful certificate generation (reason code 0), RACF may also update the DiagInfo field with the following information diagnostic data:

```
Inconsistency detected.  Signing certificate is not trusted (IRRD132I)
Inconsistency detected.  Signing certificate's date range is incorrect (IRRD113I)
```

For further information see *OS/390 SecureWay Security Server RACF Command Language Reference, SC28-1919-07* and *OS/390 SecureWay Security Server RACF Messages and Codes, SC28-1918-07*. It is expected that other security products which may be installed in place of RACF would have their own product specific diagnostic data.

9. For GENCERT, if CommonName is specified with a null value (length 0), RACF will use the PGMRNAME field from the RACF user profile as determined by the UserID field for this request. If PGMRNAME is null, the common name will be of the form of UserID:*(user's-racf-identity)*, For example: RACF UserID:JDOE. The above formula is also used if none of the subject's distinguished name fields are specified (CommonName, Title, OrgUnit, Org, Locality, StateProv, or Country).
10. For GENCERT, all CertPlist fields specified must have a non-null length except for CommonName.
11. For GENCERT, RACF forms the subject's distinguished name in the following order: CommonName, Title, OrgUnits, which is the order they appear in the CertPlist, Org, Locality, StateProv, and Country. Except as noted above, only those portions of the name specified in the CertPlist will appear in the certificate.
12. For GENCERT, OrgUnit and KeyUsage may be repeated. For all other CertPlist fields if multiple occurrences are found, the last one will be used.
13. For GENCERT, the PublicKey must be either a Netscape Navigator key, a Microsoft Internet Explorer key, or a true PKCS#10 certificate request.
14. For successful EXPORTs, the certificate returned in the CertAnchor area is a base64 encoded DER X509 certificate wrapped with the standard "-----BEGIN CERTIFICATE-----" header and "-----END CERTIFICATE-----" footer.
15. For GENCERT, no validity checking is done for the following fields: AltEmail, AltDomain, AltURI. Additionally, AltPAddr is checked form only and must be in dotted decimal form as per IP Version 4.
16. The R_PKIServ callable service creates SMF type 80 records, with event codes of 69 and 70. RACF audits the fulfillment and retrieval of digital certificate requests under the following circumstances:
    a. UAUDIT is in effect for the user
    b. The user has SPECIAL authority and SETROPTS SAUDIT is in effect
    c. The request is successful and SETROPTS AUDIT(USER) is in effect
    d. The request fails due to insufficient authorization

    For event 69, the following event qualifiers may be associated with the R_PKIServ audit records:

- 0: Successful certificate GENCERT request
- 1: Unsuccessful certificate GENCERT request due to insufficient authority

For event 70, the following event qualifiers may be associated with the R_PKIServ audit records:

- 0: Successful certificate EXPORT request
- 1: Unsuccessful certificate EXPORT request due to insufficient authority

# R_ptrace (IRRSPT00): Ptrace Authority Check

## Function

The **R_ptrace** service checks whether the calling process can ptrace the target process.

## Requirements

| | |
|---|---|
| **Authorization:** | Any PSW key in supervisor state |
| **Dispatchable unit mode:** | Task of OS/390 UNIX user |
| **Cross memory mode:** | PASN = HASN or PASN not = HASN |
| **AMODE:** | 31 |
| **RMODE:** | Any |
| **ASC mode:** | Primary or AR mode |
| **Recovery mode:** | SETFRR |
| **Serialization:** | Enabled for interrupts |
| **Locks:** | No locks held |
| **Control parameters:** | The parameter list and the work area must be in the primary address space. ALETs must be passed for all parameters except the work area. The words containing the ALETs must be in the primary address space. |

## RACF Authorization

1. **R_ptrace** checks whether the caller is a superuser or whether the caller is the owner of the target process, and verifies that the target process is not running a SETUID or SETGID program.
2. If the caller is not superuser nor the process owner, an authorization check is performed on the resource name in the UNIXPRIV class shown in Table 76. If the authorization check is successful, the caller is treated as a superuser.

*Table 76. UNIXPRIV class resource names used in R_ptrace*

| Audit function code | Resource name | Access required |
|---|---|---|
| N/A | SUPERUSER.PROCESS.PTRACE | READ |

## Format

```
CALL IRRSPT00 (Work_area,
               ALET, SAF_return_code,
               ALET, RACF_return_code,
               ALET, RACF_reason_code,
               ALET, Target_process_UIDs,
               ALET, Target_process_GIDs,
               ALET, Target_PID
              )
```

## Parameters

**Work_area**
The name of a 1024-byte work area for SAF and RACF usage. The work area must be in the primary address space.

**ALET**
The name of a word containing the ALET for the following parameter. Each parameter must have an ALET specified. Each ALET can be different. The words containing the ALETs must be in the primary address space.

**SAF_return_code**
The name of a fullword in which the SAF router returns the SAF return code.

**RACF_return_code**
The name of a fullword in which the service routine stores the return code.

**RACF_reason_code**
The name of a fullword in which the service routine stores the reason code.

**Target_process_UIDs**
The address of a 3-word area containing the real, effective, and saved OS/390 UNIX user identifiers (UIDs) (in that order) for the target process.

**Target_process_GIDs**
The address of a 3-word area containing the real, effective, and saved OS/390 UNIX group identifiers (GIDs) (in that order) for the target process.

**Target_PID**
The name of a fullword containing the PID of the target process.

## Return and Reason Codes

IRRSPT00 returns the following values in the reason and return code parameters:

| SAF Return Code | RACF Return Code | RACF Reason Code | Explanation |
|---|---|---|---|
| 0 | 0 | 0 | The service was successful. |
| 4 | 0 | 0 | RACF is not installed. |
| 8 | 8 | 4 | The caller is not authorized to ptrace the target process. |
| 8 | 8 | 12 | An internal error occurred during RACF processing. |

## Usage Notes

1. This service is intended only for use by the MVS BCP.

2. An audit record is optionally written, depending on the audit options in effect for the system.
3. This service uses task level support when OS/390 UNIX has indicated in the task's ACEE that this is a task level process.

## Related Services

None

---

# R_setegid (IRRSEG00): Set Effective GID, Set All GIDs

## Function

The **R_setegid** service checks whether the user is authorized to change the GID and, if so, changes the effective GID for the current process.

If the high-order bit of the input GID is on, the real, effective, and saved GIDs are changed for the current process.

## Requirements

| | |
|---|---|
| **Authorization:** | Any PSW key in supervisor state |
| **Dispatchable unit mode:** | Task of OS/390 UNIX user |
| **Cross memory mode:** | PASN= HASN |
| **AMODE:** | 31 |
| **RMODE:** | Any |
| **ASC mode:** | Primary or AR mode |
| **Recovery mode:** | ESTAE. Caller cannot have an FRR active. |
| **Serialization:** | Enabled for interrupts |
| **Locks:** | No locks held |
| **Control parameters:** | The parameter list and the work area must be in the primary address space. ALETs must be passed for all parameters except the work area. The words containing the ALETs must be in the primary address space. |

## RACF Authorization

1. If the high-order bit of the input GID is off and if the user is the superuser or if the input GID is equal to the real or saved GID of the calling process, the effective GID of the process is changed to the input GID. The real and saved GIDs are not changed. The new values of the GIDs are returned to the calling process.

## Format

```
CALL IRRSEG00 (Work_area,
               ALET, SAF_return_code,
               ALET, RACF_return_code,
               ALET, RACF_reason_code,
               ALET, GID,
               ALET, Output_area
              )
```

## Parameters

**Work_area**
The name of a 1024-byte work area for SAF and RACF usage. The work area must be in the primary address space.

**ALET**
The name of a word containing the ALET for the following parameter. Each parameter must have an ALET specified. Each ALET can be different. The words containing the ALETs must be in the primary address space.

**SAF_return_code**
The name of a fullword in which the SAF router returns the SAF return code.

**RACF_return_code**
The name of a fullword in which the service routine stores the return code.

**RACF_reason_code**
The name of a fullword in which the service routine stores the reason code.

**GID**
The name of a fullword containing the GID to be set. The GID must be defined to RACF. If the high-order bit is on, the GIDs stored in the output area are stored as the real, effective, and saved GIDs (in that order) for the current process.

**Output_area**
The name of a 3-word area in which the new real, effective, and saved GIDs (in that order) are returned. If the high-order bit of the GID is on, the real, effective, and saved GIDs in this area are stored as the real, effective, and saved GIDs for the current process.

## Return and Reason Codes

IRRSEG00 returns the following values in the reason and return code parameters:

| SAF Return Code | RACF Return Code | RACF Reason Code | Explanation |
|---|---|---|---|
| 0 | 0 | 0 | The service was successful. |
| 4 | 0 | 0 | RACF is not installed. |
| 8 | 8 | 4 | The GID is not defined to RACF. |
| 8 | 8 | 8 | The user is not authorized to change the GID. |
| 8 | 8 | 12 | An internal error occurred during RACF processing. |
| 8 | 8 | 16 | Recovery could not be established. |

## Usage Notes

1. This service is intended only for use by the MVS BCP.
2. IRRSEG00 only changes the GIDs. The user's current group in the ACEE is not changed. Therefore, IRRSEG00 only affects access to OS/390 UNIX files. Access to other MVS files is not changed.
3. An audit record is written.

## Related Services

None

# R_seteuid (IRRSEU00): Set Effective UID, Set All UIDs

## Function

The **R_seteuid** service checks whether the user is authorized to change the OS/390 UNIX user identifiers (UIDs) and, if so, changes the effective UID for the current process.

If the high-order bit of the input UID is on, the real, effective, and saved UIDs are changed for the current process.

## Requirements

| | |
|---|---|
| **Authorization:** | Any PSW key in supervisor state |
| **Dispatchable unit mode:** | Task of OS/390 UNIX user |
| **Cross memory mode:** | PASN= HASN |
| **AMODE:** | 31 |
| **RMODE:** | Any |
| **ASC mode:** | Primary or AR mode |
| **Recovery mode:** | ESTAE. Caller cannot have an FRR active. |
| **Serialization:** | Enabled for interrupts |
| **Locks:** | No locks held |
| **Control parameters:** | The parameter list and the work area must be in the primary address space. ALETs must be passed for all parameters except the work area. The words containing the ALETs must be in the primary address space. |

## RACF Authorization

1. If the high-order bit of the input OS/390 UNIX user identifier (UID) is off and if the user is the superuser or if the input UID is equal to the real or saved UID of the calling process, the effective UID of the process is changed to the input UID. The real and saved UIDs are not changed. The new values of the UIDs are returned to the calling process.

## Format

```
CALL IRRSEU00 (Work_area,
               ALET, SAF_return_code,
               ALET, RACF_return_code,
               ALET, RACF_reason_code,
               ALET, UID,
               ALET, Output_area
              )
```

## Parameters

**Work_area**
The name of a 1024-byte work area for SAF and RACF usage. The work area must be in the primary address space.

**ALET**
The name of a word containing the ALET for the following parameter. Each parameter must have an ALET specified. Each ALET can be different. The words containing the ALETs must be in the primary address space.

**SAF_return_code**
The name of a fullword in which the SAF router returns the SAF return code.

**RACF_return_code**
The name of a fullword in which the service routine stores the return code.

**RACF_reason_code**
The name of a fullword in which the service routine stores the reason code.

**UID**
The name of a fullword containing the OS/390 UNIX user identifier (UID) to be set. The UID must be defined to RACF. If the high-order bit is on, the UIDs stored in the output area are stored as the real, effective, and saved UIDs (in that order) for the current process.

**Output_area**
The name of a 3-word area in which the new real, effective, and saved OS/390 UNIX user identifiers (UIDs) (in that order) are returned. If the high-order bit of the UID is on, the real, effective, and saved UIDs in this area are stored as the real, effective, and saved UIDs for the current process.

## Return and Reason Codes

IRRSEU00 returns the following values in the reason and return code parameters:

| SAF Return Code | RACF Return Code | RACF Reason Code | Explanation |
|---|---|---|---|
| 0 | 0 | 0 | The service was successful. |
| 4 | 0 | 0 | RACF is not installed. |
| 8 | 8 | 4 | The OS/390 UNIX user identifier (UID) is not defined to RACF. |
| 8 | 8 | 8 | The user is not authorized to change the OS/390 UNIX user identifier (UID). |
| 8 | 8 | 12 | An internal error occurred during RACF processing |
| 8 | 8 | 16 | Recovery could not be established. |

## Usage Notes

1. This service is intended only for use by the MVS BCP.
2. For additional security-related information, see the description of the **seteuid** callable service in *OS/390 UNIX System Services Programming: Assembler Callable Services Reference*
3. An audit record is written.

## Related Services

None

# R_setgid (IRRSSG00): Set Group Name

## Function

The **R_setgid** service checks whether the user is authorized to change the GIDs and, if so, changes the real, saved, or effective GID (or some combination of these) for the current process.

## Requirements

| | |
|---|---|
| **Authorization:** | Any PSW key in supervisor state |
| **Dispatchable unit mode:** | Task of OS/390 UNIX user |
| **Cross memory mode:** | PASN = HASN |
| **AMODE:** | 31 |
| **RMODE:** | Any |
| **ASC mode:** | Primary or AR mode |
| **Recovery mode:** | ESTAE. Caller cannot have an FRR active. |
| **Serialization:** | Enabled for interrupts |
| **Locks:** | No locks held |
| **Control parameters:** | The parameter list and the work area must be in the primary address space. ALETs must be passed for all parameters except the work area. The words containing the ALETs must be in the primary address space. |

## RACF Authorization

1. If the calling process is a superuser, the real, saved, and effective GIDs are changed. If the calling process is not a superuser but the input GID is equal to the real or saved GID, the process's effective GID is changed. If neither condition is met, the process's GIDs are not changed, and an error return code and an error reason code are returned.

## Format

```
CALL IRRSSG00 (Work_area,
               ALET, SAF_return_code,
               ALET, RACF_return_code,
               ALET, RACF_reason_code,
               ALET, GID,
               ALET, Output_area
              )
```

## Parameters

**Work_area**
The name of a 1024-byte work area for SAF and RACF usage. The work area must be in the primary address space.

**ALET**
The name of a word containing the ALET for the following parameter. Each parameter must have an ALET specified. Each ALET can be different. The words containing the ALETs must be in the primary address space.

**SAF_return_code**
The name of a fullword in which the SAF router returns the SAF return code.

**RACF_return_code**
The name of a fullword in which the service routine stores the return code.

**RACF_reason_code**
The name of a fullword in which the service routine stores the reason code.

**GID**
The name of a fullword containing the GID to be set. The GID must be defined to RACF.

**Output_area**
The name of a 3-word area in which the new real, effective, and saved GIDs (in that order) are returned.

## Return and Reason Codes

IRRSSG00 returns the following values in the reason and return code parameters:

| SAF Return Code | RACF Return Code | RACF Reason Code | Explanation |
|---|---|---|---|
| 0 | 0 | 0 | The service was successful. |
| 4 | 0 | 0 | RACF is not installed. |
| 8 | 8 | 4 | The GID is not defined to RACF. |
| 8 | 8 | 8 | The user is not authorized to change the GID. |
| 8 | 8 | 12 | An internal error occurred during RACF processing. |
| 8 | 8 | 16 | Recovery could not be established. |

## Usage Notes

1. This service is intended only for use by the MVS BCP.
2. IRRSSG00 changes only the GIDs. No change is made to the user's current group in the ACEE. Therefore, IRRSSG00 only affects access to OS/390 UNIX files. Access to other MVS files is unchanged.
3. An audit record is written.

## Related Services

None

# R_setuid (IRRSSU00): Set OS/390 UNIX user identifier (UID)

## Function

The **R_setuid** service checks whether the user is authorized to change the OS/390 UNIX user identifiers (UIDs) and, if so, changes the real, saved, or effective UID (or some combination of these) for the current process.

## Requirements

**Authorization:** Any PSW key in supervisor state
**Dispatchable unit mode:** Task of OS/390 UNIX user
**Cross memory mode:** PASN = HASN
**AMODE:** 31
**RMODE:** Any
**ASC mode:** Primary or AR mode
**Recovery mode:** ESTAE. Caller cannot have an FRR active.
**Serialization:** Enabled for interrupts
**Locks:** No locks held

**Control parameters:**

The parameter list and the work area must be in the primary address space. ALETs must be passed for all parameters except the work area. The words containing the ALETs must be in the primary address space.

## RACF Authorization

1. If the calling process is a superuser, the real, saved, and effective OS/390 UNIX user identifiers (UIDs) are changed. If the calling process is not a superuser, but the input UID is equal to the real or saved UID, the process's effective UID is changed. If neither condition is met, the process's UIDs are not changed, and an error return code and and an error reason code are returned.

## Format

```
CALL IRRSSU00 (Work_area,
               ALET,SAF_return_code,
               ALET, RACF_return_code,
               ALET, RACF_reason_code,
               ALET, UID,
               ALET, Output_area
              )
```

## Parameters

**Work_area**
The name of a 1024-byte work area for SAF and RACF usage. The work area must be in the primary address space.

**ALET**
The name of a word containing the ALET for the following parameter. Each parameter must have an ALET specified. Each ALET can be different. The words containing the ALETs must be in the primary address space.

**SAF_return_code**
The name of a fullword in which the SAF router returns the SAF return code.

**RACF_return_code**
The name of a fullword in which the service routine stores the return code.

**RACF_reason_code**
The name of a fullword in which the service routine stores the reason code.

**UID**
The name of a fullword containing the OS/390 UNIX user identifier (UID) to be set. The UID must be defined to RACF.

**Output_area**
The name of a 3-word area in which the new real, effective, and saved OS/390 UNIX user identifiers (UIDs) (in that order) are returned.

## Return and Reason Codes

IRRSSU00 returns the following values in the reason and return code parameters:

| SAF Return Code | RACF Return Code | RACF Reason Code | Explanation |
|---|---|---|---|
| 0 | 0 | 0 | The service was successful. |

| SAF Return Code | RACF Return Code | RACF Reason Code | Explanation |
|---|---|---|---|
| 4 | 0 | 0 | RACF is not installed. |
| 8 | 8 | 4 | The OS/390 UNIX user identifier (UID) is not defined to RACF. |
| 8 | 8 | 8 | The user is not authorized to change the OS/390 UNIX user identifier (UID). |
| 8 | 8 | 12 | An internal error occurred during RACF processing. |
| 8 | 8 | 16 | Recovery could not be established. |

## Usage Notes

1. This service is intended only for use by the MVS BCP.
2. For additional security-related information, see the description of the **setuid** callable service in *OS/390 UNIX System Services Programming: Assembler Callable Services Reference*.
3. An audit record is optionally written, depending on the audit options in effect for the system.

## Related Services

None

# R_ticketserv (IRRSPK00): Parse or Extract

## Function

The **R_ticketserv** service enables OS/390 application servers to parse or extract principal names from a GSS-API context token. This enables an OS/390 application server to determine the client principal who originated an application-specific request, when the request includes a GSS-API context token and the intended recipient is the OS/390 application server. For more information on the GSS-API services supported on OS/390, see the SecureWay Security Server Network Authentication and Privacy Service product documentation.

## Requirements

**Authorization:** Any PSW key in supervisor state or problem state

**Dispatchable unit mode:** Task of user

**Cross memory mode:** PASN = HASN

**AMODE:** 31

**RMODE:** Any

**ASC mode:** Primary or AR mode

**Recovery mode:** ESTAE. Caller cannot have an FRR active.

**Serialization:** Enabled for interrupts

**Locks:** No locks held

**Control parameters:** The parameter list and the work area must be in the primary address space. The words containing the ALETs must be in the primary address space.

## Linkage Conventions

The parameter list for this callable service is intended to be variable length to allow for future expansion. To allow for this, the last word in the parameter list must have a one in the high-order (sign) bit.

## RACF Authorization

For servers not running in system key or supervisor state, the use of R_ticketserv service is authorized by the resource IRR.RTICKETSERV in the FACILITY class. The application server must be running with a RACF user or group ID that has at least READ authority to this resource. If the class is inactive, or the resource is not defined, only servers running system key or supervisor state may use the R_ticketserv service.

## Format

```
CALL IRRSPK00 (Work_area,
               ALET, SAF_return_code,
               ALET, RACF_return_code,
               ALET, RACF_reason_code,
               ALET, Function_code,
                        Option_word,
                        Ticket_area,
                        Ticket_options,
                        Ticket_principal_userid,
              )
```

## Parameters

**Work_area**
The name of a 1024-byte work area for SAF and RACF usage. The work area must be in the primary address space.

**ALET**
The name of a word containing the ALET for the following parameter. Each parameter must have an ALET specified. Each ALET can be different. The words containing the ALETs must be in the primary address space.

**SAF_return_code**
The name of a full word in which the SAF router returns the SAF return code.

**RACF_return_code**
The name of a full word in which the service routine stores the return code.

**RACF_reason_code**
The name of a full word in which the service routine stores the reason code.

**ALET**
The name of a word which must be in the primary address space and which contains the ALET for the following fields:
- Function_code
- Option_word
- Ticket_area
- Ticket_options
- Ticket_principal_userid

**Function_code**
The name of a half word (2 byte) area containing the Function code. The function code has one of the following values:

**X'0001'**
Parse specified ticket and return Network Authentication and Privacy Service principal name.

**Option_word**
The name of a fullword containing binary zeros. The area pointed to by this parameter is reserved for future use.

**Ticket_area**
The name of an area that consists of a 2 byte field, followed by the GSS-API context token. For function code 1, extract Network Authentication and Privacy Service V5 principal, the GSS-API context token is assumed by SAF to have been created exclusively by the Network Authentication and Privacy Service mechanism.

**Ticket_options**
The address of a binary bit string which identifies the ticket specific processing to be performed. This parameter is reserved for future use.

**Ticket_principal_userid**
The name of a 242 byte area that consists of a 2 byte length field followed by the name of the Ticket principal user ID.

**Note:** Fully qualified Network Authentication and Privacy Service names will be returned, using a case sensitive, DCE-like naming convention: /.../realm_name/principal_name

## Return and Reason Codes

IRRSPK00 may return the following values in the reason and return code parameters:

| SAF Return Code | RACF Return Code | RACF Reason Code | Explanation |
|---|---|---|---|
| 0 | 0 | 0 | The service was successful. |
| 4 | 0 | 0 | RACF is not installed. |
| 8 | 8 | 0 | Invalid function code. |
| 8 | 8 | 4 | Parameter list error. |
| 8 | 8 | 9 | An internal error was encountered. |
| 8 | 8 | 12 | A recovery environment could not be established. |
| 8 | 8 | 16 | Not authorized to use this service. |
| 8 | 12 | 8 | Invocation of the SecureWay Security Server Network Authentication and Privacy Service Program Call (PC) interface failed with a 'parameter buffer overflow' return code. This indicates an internal error in IRRSPK00. |
| 8 | 12 | 12 | Invocation of the SecureWay Security Server Network Authentication and Privacy Service Program Call (PC) interface failed with an 'unable to allocate storage' return code. The region size for the SecureWay Security Server Network Authentication and Privacy Service started task (SKRBKDC) should be increased. |

| SAF Return Code | RACF Return Code | RACF Reason Code | Explanation |
|---|---|---|---|
| 8 | 12 | 16 | Invocation of the SecureWay Security Server Network Authentication and Privacy Service Program Call (PC) interface failed with a 'local services are not available' return code. This indicates that the SecureWay Security Server Network Authentication and Privacy Service started task (SKRBKDC) address space has not been started or is terminating. |
| 8 | 12 | 20 | Invocation of the SecureWay Security Server Network Authentication and Privacy Service Program Call (PC) interface failed with a 'abend in the PC service routine' return code. The symptom record associated with this abend can be found in the logrec dataset. |
| 8 | 12 | 24 | Invocation of the SecureWay Security Server Network Authentication and Privacy Service Program Call (PC) interface failed with an 'unable to obtain control lock' return code. This can occur if the task holding the lock is not being dispatched (for example, a dump is in progress). |
| 8 | 16 | X'nnnnnnnn' | The SecureWay Security Server Network Authentication and Privacy Service was not able to successfully extract the client principal name from the supplied Kerberos V5 ticket. X'nnnnnnnn' is the Kerberos return code. Refer to the SecureWay Security Server Network Authentication and Privacy Service documentation for more information. |

## Usage Notes

1. This service is intended for use by OS/390 application servers. It allows application servers with a Kerberos V5 ticket to determine the Kerberos principal associated with the ticket.
2. This service requires that the SecureWay Security Server Network Authentication and Privacy Service be installed and executing. Otherwise, SAF return code 8, RACF return code 12, and RACF reason code 16 will be returned to the invoker.
3. In a datasharing sysplex, there must be an SecureWay Security Server Network Authentication and Privacy Service instance running on each system in the sysplex. The SecureWay Security Server Network Authentication and Privacy Service instances must all be in the same realm and share the same RACF database (if they do not share the same database, then they cannot be in the same realm).
4. An ALET must be specified for the SAF_return_code, RACF_return_code, and RACF_reason_code parameters, and a single ALET specified for all of the remaining paramenters.

5. The parameter list for this callable service is intended to be variable length to allow for future expansion. To allow for this, the last word in the parameter list much have a one in the high-order (sign) bit. If the last word in the parameter list does not have a one in the high-order (sign) bit, the caller receives a parameter list error. The first parameter that can have the high-order bit on, ending the parameter list, is the Ticket_principal_userid parameter.
6. The calling OS/390 application server must have a local Kerberos identity defined as a populated KERB segment of the server's RACF User profile.
7. For function code X'0001', a SAF return code 8 and a RACF return code 16 indicates that the SecureWay Security Server Network Authentication and Privacy Service was unable to successfully process the input Kerberos V5 ticket. The return code is passed back to the invoker as the RACF reason code. Some of the more common return codes are the following:
   - X'861B6D04' (G_BUFFER_ALLOC)=storage not available for GSS-API control block.
   - X'861B6D06' (G_WRONG_SIZE)=client principal name is too long for result buffer.
   - X'861B6D0B' (G_BAD_TOK_HEADER)=the GSS-API token header is incorrect.
   - X'861B6D58' (G_UNEXPECTED_TOKEN)=the GSS-API token was not created by the gss_init_sec_context() function.
   - X'861B6D60' (G_UNSUPPORTED_MECHANISM)=unsupported GSS-API security mechanism.
   - X'96C73A07'(KRB5KDC_ERR_S_PRINCIPAL_UNKNOWN)=the current RACF userid is not associated with a Kerberos principal.
   - X'96C73A20'(KRB5KDC_AP_ERR_TKT_EXPIRED)=Kerberos ticket is expired.
   - X'96C73A25'(KRB5KDC_AP_ERR_SKEW)=Client and server clocks are not synchronized or authenticator is expired.
   - X'96C73A90'(KRB5KDC_AP_WRONG_PRINC)=the server principal in the GSS-API security token does not match the principal associated with the current RACF userid.
   - X'96C73C02'(KRB5_NOMEM)=storage not available for Kerberos control block.

## Parameter Usage

| Parameter | Function Code X'0001' |
|---|---|
| SAF_return_code | Output |
| RACF_return_code | Output |
| RACF_reason_code | Output |
| Function _code | Input |
| Option_word | Reserved |
| Ticket_area | Input |
| Ticket_options | Reserved |
| Ticket_principal_userid | Output |

## Related Services

R_kerbinfo, R_usermap

# R_umask (IRRSMM00): Set File Mode Creation Mask

## Function

The **R_umask** service sets the file mode creation mask for the current process to the permission bits specified in the input mode parameter. It returns the permission bits that were in the file mode creation map in the mode parameter.

## Requirements

| | |
|---|---|
| **Authorization:** | Any PSW key in supervisor state |
| **Dispatchable unit mode:** | Task of OS/390 UNIX user |
| **Cross memory mode:** | PASN = HASN or PASN not = HASN |
| **AMODE:** | 31 |
| **RMODE:** | Any |
| **ASC mode:** | Primary or AR mode |
| **Recovery mode:** | None |
| **Serialization:** | Enabled for interrupts |
| **Locks:** | No locks held |
| **Control parameters:** | The parameter list and the work area must be in the primary address space. ALETs must be passed for all parameters except the work area. The words containing the ALETs must be in the primary address space. |

## RACF Authorization

None

## Format

```
CALL IRRSMM00 (Work_area,
               ALET, SAF_return_code,
               ALET, RACF_return_code,
               ALET, RACF_reason_code,
               ALET, Mode
              )
```

## Parameters

**Work_area**
The name of a 1024-byte work area for SAF and RACF usage. The work area must be in the primary address space.

**ALET**
The name of a word containing the ALET for the following parameter. Each parameter must have an ALET specified. Each ALET can be different. The words containing the ALETs must be in the primary address space.

**SAF_return_code**
The name of a fullword in which the SAF router returns the SAF return code.

**RACF_return_code**
The name of a fullword in which the service routine stores the return code.

**RACF_reason_code**
The name of a fullword in which the service routine stores the reason code.

**Mode**
The name of a word containing the mode bits to be set in file mode creation

mask of the current process. Only the file permission bits in the mode parameter are used. Other defined bits are ignored.

On output, the mode word is zeroed and the permission bits that were in the file mode creation mask are set in the mode word.

See "File Type and File Mode Values" on page 4 for a definition of the security bits in the mode parameter.

## Return and Reason Codes

IRRSMM00 returns the following values in the reason and return code parameters:

| SAF Return Code | RACF Return Code | RACF Reason Code | Explanation |
|---|---|---|---|
| 0 | 0 | 0 | The service was successful. |
| 4 | 0 | 0 | RACF is not installed. |

## Usage Note

- This service is intended only for use by the MVS BCP and by OS/390 UNIX System Services servers. The service contains support for OS/390 UNIX System Services servers, but can not be directly invoked by an OS/390 UNIX System Services server.

## Related services

chmod, makeFSP

---

# R_usermap (IRRSIM00): Map application user

## Function

The **R_usermap** service enables OS/390 application servers to determine the application user identity associated with a RACF user ID, or to determine the RACF user ID associated with an application user identity or digital certificate. Examples of application user identities suported are Lotus Notes for OS/390 and Novell Directory Services (NDS).

This service can only map application user identities which have already been defined to RACF:

- For Lotus Notes for OS/390, the RACF USER profile must have an LNOTES segment containing a short name. This can be added with the ADDUSER or ALTUSER command, or the R_admin callable service.
- For NDS for OS/390, the RACF USER profile must have an NDS segment containing a user name. This can be added with the ADDUSER or ALTUSER command, or the R_admin callable service.
- For digital certificates, the certificate must be associated with a RACF user ID through automatic registration or with the RACDCERT command.
- For SecureWay Security Server Network Authentication and Privacy Service, local Kerberos principals require a RACF USER profile with a KERB segment containing a principal name. Foreign Kerberos principals must be defined to RACF using KERBLINK profiles.

## Requirements

| | |
|---|---|
| **Authorization:** | Any PSW key in supervisor or problem state |
| **Dispatchable unit mode:** | Task of user |
| **Cross memory mode:** | PASN = HASN |
| **AMODE:** | 31 |
| **RMODE:** | Any |
| **ASC mode:** | Primary or AR mode |
| **Recovery mode:** | ESTAE. Caller cannot have an FRR active. |
| **Serialization:** | Enabled for interrupts |
| **Locks:** | No locks held |
| **Control parameters:** | The parameter list and the work area must be in the primary address space. The words containing the ALETs must be in the primary address space. |

## Linkage Conventions

The parameter list for this callable service is intended to be variable length to allow for future expansion. To allow for this, the last word in the parameter list must have a 1 in the high-order (sign) bit.

## RACF Authorization

For servers not running in system key or supervisor state, the use of the R_usermap service is authorized by the resource **IRR.RUSERMAP** in the **FACILITY** class. The application server must be running with a RACF user ID or group ID that has at least **READ** authority to this resource. If the class is inactive, or the resource is not defined, only servers running in system key or supervisor state may use the R_usermap service.

## Format

```
CALL IRRSIM00 (Work_area,
               ALET, SAF_return_code,
               ALET, RACF_return_code,
               ALET, RACF_reason_code,
               ALET, (Function_code,
                        Option_word,
                        RACF_userid,
                        Certificate,
                        Application_userid
               )
```

## Parameters

**Work_area**
The name of a 1024-byte work area for SAF and RACF usage. The work area must be in the primary address space and must be on a double word boundary.

**ALET**
The name of a word containing the ALET for the following parameter. Each ALET can be different. The last ALET in the parameter list will be used for the remainder of the parameters. The words containing the ALETs must be in the primary address space.

**SAF_return_code**
The name of a fullword in which the SAF router returns the SAF return code.

**RACF_return_code**
The name of a fullword in which the service routine stores the return code.

**RACF_reason_code**
The name of a fullword in which the service routine stores the reason code.

**Function_code**
The name of a halfword containing the function code. The function code has one of the following values:

**X'0001'**
Return the Lotus Notes for OS/390 application user identity associated with the supplied RACF user ID.

**X'0002'**
Return the RACF user ID associated with the supplied Lotus Notes for OS/390 application user identity or digital certificate.

**X'0003'**
Return the NDS for OS/390 application user identity associated with the supplied RACF user ID.

**X'0004'**
Return the RACF user ID associated with the supplied NDS for OS/390 application user identity or digital certificate.

**X'0005'**
Return the Network Authentication and Privacy Service application user identity associated with the supplied RACF user ID.

**Note:** This functions only with local Network Authentication and Privacy Service principals.

**X'0006'**
Return the RACF user ID associated with the supplied Network Authentication and Privacy Service application user identity or digital certificate.

**Option_word**
The name of a fullword containing binary zeros. The area pointed to by this parameter is reserved for future use.

**RACF_userid**
The name of a 9 byte area that consists of a 1 byte length field followed by up to 8 characters. It must be specified in upper case. If not specified, the length must equal 0.

**Certificate**
The name of an area that consists of a 4 byte length field followed by a digital certificate. The certificate must be a single BER encoded X.509 certificate. If not specified, the length must equal 0.

**Application_userid**
The name of a 248 byte area that consists of a 2 byte length field followed by the name of the application user identity. If not specified, the length must equal 0.

## Return and Reason Codes

R_usermap may return the following values in the reason and return code parameters:

| SAF return code | RACF return code | RACF reason code | Explanation |
|---|---|---|---|
| 0 | 0 | 0 | The service was successful. |
| 4 | 0 | 0 | RACF is not installed. |
| 8 | 8 | 4 | Parameter list error occurred. |
| 8 | 8 | 8 | An internal error occurred during RACF processing. |
| 8 | 8 | 12 | Recovery environment could not be established. |
| 8 | 8 | 16 | There is no mapping between RACF and an application. For function codes 1 and 3, and 5, the RACF user ID exists but there is either no SNAME in the LNOTES segment or no LNOTES segment, or there is no UNAME in the NDS segment or no NDS segment, or no KERBNAME in the KERB segment, or. no KERB segment. For function codes 2, 4, and 6, there is no mapping to a RACF user ID for the application identity provided. |
| 8 | 8 | 20 | Not authorized to use this service. |
| 8 | 8 | 24 | The specified RACF user ID does not exist. |
| 8 | 8 | 28 | Certificate is not valid. |
| 8 | 8 | 32 | Either no RACF user ID is defined for this certificate, or the certificate status is NOTRUST. |

## Parameter Usage

*Table 77. Parameter Usage*

| Parameter | Function Code 1 (RACF to Notes) | Function Code 2 (Notes to RACF) | Function Code 3 (RACF to NDS) | Function Code 4 (NDS to RACF) | Function Code 5 (RACF to KERB) | Function Code 6 (RACF to KERB) |
|---|---|---|---|---|---|---|
| SAF_return_code | Output | Output | Output | Output | Output | Output |
| RACF_return_code | Output | Output | Output | Output | Output | Output |
| RACF_reason_code | Output | Output | Output | Output | Output | Output |
| Function_code | Input | Input | Input | Input | Input | Input |
| Option_word | Reserved | Reserved | Reserved | Reserved | Reserved | Reserved |
| RACF_userid | Input | Output | Input | Output | Input | Output |
| Certificate | N/A | Input | N/A | Input | N/A | Input |
| Application_userid | Output | Input | Output | Input | Output | Input |

## Usage Notes

1. This service is intended for use by OS/390 application servers. It allows them to map between supported application user identities and the corresponding RACF user ID, or to determine the RACF user ID by supplying the corresponding application user identity or digital certificate.
2. An **ALET** must be specified for the SAF_return_code, RACF_return_code, RACF_reason_code parameters, and a single **ALET** specified for all of the remaining parameters.
3. The parameter list for this callable service is intended to be variable length to allow for future expansion. To allow for this, the last word in the parameter list must have a 1 in the high-order bit. If the last word in the parameter list does not have a 1 in the high-order (sign) bit, the caller receives a parameter list error. The first parameter that can have the high-order bit on, ending the parameter list, is the Application_userid parameter.
4. If the function_code indicates that an application identity is to be returned, and no RACF_userid is supplied, the caller receives a parameter list error.
5. If the function_code indicates that a RACF user ID is to be returned, and no application_userid or certificate is supplied, the caller receives a parameter list error.
6. Specification of an unknown function code, a RACF_userid with a length greater than 8 or an application_userid with a length greater than 246 will result in a parameter list error.
7. If the function_code indicates that a RACF user ID is to be returned, and both an application_userid and a certificate are supplied, the application_userid will be used.
8. If the function_code indicates that an application user identity is to be returned, the caller is expected to supply a 248 byte area for the application_userid parameter. If an application user identity is defined for the RACF_userid specified, R_usermap will update this area with the length and value of the application user identity. This storage must be accessible in the caller's key.
9. If the function_code indicates that a RACF user ID is to be returned, the caller is expected to supply a 9 byte area for the RACF_userid parameter. If a RACF user ID is associated with the application_identity specified, R_usermap will update this area with the length and value of the RACF user ID. This storage must be accessible in the caller's key.
10. The conversion between the supported application user identities (or certificates) and RACF user IDs is dependent on the definition of the application specific segments associated with the RACF USER profile. To convert between Lotus Notes for OS/390 user identity and a RACF user ID, the RACF USER profile must have an LNOTES segment containing the SNAME field. To convert between an NDS for OS/390 user identity and a RACF user ID, the RACF USER profile must have an NDS segment containing the UNAME field. To convert between a certificate and a RACF user ID, the RACF USER profile must be associated with the certificate.

    **Note:** In the case of SecureWay Security Server Network Authentication and Privacy Service identities, local SecureWay Security Server Network Authentication and Privacy Service principals are similar to the above: to convert between a SecureWay Security Server Network Authentication and Privacy Service local principal identity and a RACF user ID, the RACF USER profile must have a KERB segment containing the KERBNAME field. However, in the case of foreign SecureWay

Security Server Network Authentication and Privacy Service principals, a KERBLINK class profile must be defined to map the foreign SecureWay Security Server Network Authentication and Privacy Service principal to a RACF user identity. The RACF user identity associated with a foreign SecureWay Security Server Network Authentication and Privacy Service principal (or multiple foreign SecureWay Security Server Network Authentication and Privacy Service principals), does not require a KERB segment.

11. The certificate supplied by the certificate parameter is used only to identify a RACF user ID. It is expected that the certificate was previously verified. Note the following additional details regarding certificate processing:
    a. All fields as defined for X.509 version 1 certificates must be present and non-null.
    b. X.509 certificates with version numbers greater than 3 are not supported.
    c. Version 3 certificates with critical extensions are not supported. Noncritical extensions are ignored.
    d. Subject and issuer names can contain only the following string types:
       - T61STRING - TAG 20
       - PRINTABLESTRING - TAG 19
       - IA5STRING - TAG 22
       - VISIBLESTRING - TAG 26
       - GENERALSTRING - TAG 27
    e. The length of the serial number plus the length of the issuer's name cannot exceed 245.
    f. No date validity check is performed on the certificate.
    g. No signature check is performed on the certificate.
12. If the certificate supplied by the caller is defined to RACF with a status of NOTRUST, R_usermap will return a RACF return code 8, RACF reason code 32, indicating that no user ID is defined to use this certificate.
13. For function_codes 5 and 6, the application user identity is the SecureWay Security Server Network Authentication and Privacy Service principal name. Local principal name is case sensitive, foreign principal is not (the RDEFINE of a KERBLINK profile will fold the name to upper case). R_usermap will accept mixed case input of foreign profile names, but will fold to upper case before attempting to locate the appropriate KERBLINK identity mapping profile. R_usermap output of foreign principal names will always be upper case. Additionally, while local principal names may be supplied fully qualified with the name of the local realm, R_usermap output of local principal names will always be unqualified. Realm qualified names follow a DCE-like convention of /.../realm_name/principal_name.
14. If the function_code specifies that a RACF user ID is to be returned and the length supplied for the Application_user is greater than the maximum allowed, such as greater than 64 for a Lotus Notes for OS/390 user identity, or greater than 240 for SecureWay Security Server Network Authentication and Privacy Service user identity, the caller receives the ″no mapping between RACF and an application″ error.
15. Check if any logrec entry has been created to ensure R_usermap service was being run successfully and also refer to *OS/390 SecureWay Security Server RACF Diagnosis Guide* for detailed logrec information.

## Related Services

R_dceinfo, R_dceruid

# Chapter 3. IRRSXT00 Installation Exit

IRRSXT00 installation exit uses the IRRSXT00 module as described in this chapter.

## Function

As described in "Chapter 1. Using the RACF Callable Services" on page 1, Linkage Conventions for the Callable Services, IRRSXT00 is invoked by the SAF callable services router before and after RACF is called. It receives as input, a function code indicating which callable service is being called, and the parameter list that will be passed to RACF. The first parameter in the parameter list points to a work area. IRRSXT00 can use the first 152 bytes of this work area. The first word of the work area is set to zero before the pre-RACF call to the exit. IRRSXT00 should set another value in this word to indicate to the post-RACF exit call that it is the second call. The first four words of the work area are passed unchanged from the pre-RACF to the post-RACF exit.

The pre-RACF exit can change the content of the parameter list that will be passed to the external security product. It can also indicate with return codes that the external security product should be bypassed and control returned to the caller. The SAF return code is set based on the exit return code. If the external security product is bypassed, the exit routine must provide all of the output including RACF-compatible return and reason codes that the invokers of the services expect.

The post-RACF exit can look at or change the output from RACF including the RACF return and reason codes. No exit return codes are defined from this exit call. SAF return codes are set based on the RACF return codes, not on an exit return code.

## Requirements

| | |
|---|---|
| **Authorization:** | PSW key 0 in supervisor state |
| **Dispatchable unit mode:** | Task of user calling security function |
| **Cross memory mode:** | PASN = HASN or PASN not = HASN |
| **AMODE:** | Any *or* 31 |
| **RMODE:** | Any *or* 24 |
| **ASC mode:** | AR mode |
| **Serialization:** | Enabled for interrupts |
| **Locks:** | No locks held |
| **Control parameters:** | The parameter list and the work area are in the primary address space. ALETs are passed for all parameters except the work area. The words containing the ALETs are in the primary address space. |

## Interface Registers

| Register | AR content | GR content |
|---|---|---|
| **Input Registers** | | |

| | | |
|---|---|---|
| 0 | Any | Function code of service. Refer to the description of IRRPFC in *OS/390 SecureWay Security Server RACF Data Areas*. |
| 1 | 0 | Address of parameter list |
| 2 - 13 | Any | Undefined |
| 14 | 0 | Return address |
| 15 | 0 | Entry point address |
| **Output Registers** | | |
| 0 - 14 | Same as input | Same as input |
| 15 | Undefined | Return code |

## Input

The parameter list is the same as the list that will be passed to the external security product (for example, RACF). The content of the list varies depending on the service requested. See *OS/390 SecureWay Security Server RACF Data Areas* for descriptions of the parameter list, IRRPCOMP, as well as detailed descriptions of structures such as the FSP and CRED, which are passed as parameters on a number of the callable services. See "Chapter 2. Callable Services Descriptions" on page 9 for descriptions of the possible parameter lists.

The first 152 bytes of the work area pointed to by the parameter list can be used by the exit. The rest of the work area is reserved for SAF and RACF. The first four words of the 152-byte area can be used to pass data from the pre-RACF call of the exit to the post-RACF call to the exit. These words are not used by SAF or RACF. The other 136 bytes may be used by RACF services or the SAF router between the calls to the pre-RACF exit and the post-RACF exit.

## Output

**Returned Data:** If the exit indicates that RACF is not to be called, the exit is responsible for setting the RACF return and reason codes and providing any other output data expected by the caller of the requested service.

Refer to "Chapter 2. Callable Services Descriptions" on page 9 for information on which callable services return output to their callers, as well as the format of the output.

**Note:** The return and reason codes are not stored in the parameter list as they are for SAF exit ICHRTX00. For IRRSXT00, the parameter list contains the addresses of two words in which the return and reason codes must be stored.

The pre-RACF exit routine must restore all registers on return except register 15, which must contain a return code.

The post-RACF exit must also restore all registers except 15. No return codes are defined for the post-RACF exit.

**Return Codes:** The pre-RACF exit can return one of the following return codes:

| Return code | Explanation |
|---|---|
| 0 | Exit complete - continue processing and call RACF for further security processing. The exit routine may change the content of the parameter list that will be passed to RACF |
| 200 | Exit complete - access authorized. The SAF callable services router sets SAF router return code 0 and returns to the caller of the service, bypassing any further security processing. |
| 204 | Exit complete - no decision. The SAF callable services router sets SAF return code 4 and returns to the caller of the service, bypassing any further security processing. |
| 208 | Exit complete - access not authorized. The SAF callable services router sets SAF return code 8 and returns to the caller of the service, bypassing any further security processing. |
| Other | Exit complete - the SAF callable services router sets the SAF return code to the exit-supplied value and returns to the caller of the service, bypassing any further security processing. |

## Usage Notes

1. IRRSXT00 must be reentrant.
2. IRRSXT00 can receive control in cross memory mode.
3. IRRSXT00 must use BAKR to save registers. No save area is provided on entry. It is called in AR mode and must save both general and access registers.
4. To install the SAF callable services router installation exit, create the load module, name it IRRSXT00, and load it into the link pack area (LPA).
5. IRRSXT00 must provide its own recovery routine. If the exit routine terminates abnormally, its recovery routine gets control. If a recovery routine is not provided or if the recovery routine percolates, the recovery routine of the caller of the service stub will get control.
6. To determine the caller of the SAF callable service, use the function code in register 0 to determine which callable service is being called. Refer to Table 1 on page 9, in Chapter 2, for the list of components and products that are possible callers of each service.

   The usage notes that follow the callable service descriptions are also helpful in determining the caller.

   Many services receive the CRED as a parameter. The CRED contains an audit function code, defined in macro IRRPAFC, which identifies the OS/390 UNIX function calling the service. Additional information on which callable services are called by OS/390 UNIX functions can be found in the OS/390 Security Server (RACF) Auditor's Guide, under Classes that Control Auditing for OS/390 UNIX System Services.

# Appendix. Notices

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

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

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

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
USA

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

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

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

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

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

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
Mail Station P300
2455 South Road
Poughkeepsie, NY 12601-5400
USA

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

This product contains code licensed from RSA Data Security Incorporated.


## Programming Interface Information

This manual is intended to describe the RACF callable services. This publication documents intended Programming Interfaces that allow an installation to write programs to obtain the services of RACF.

## Trademarks

The following terms are trademarks of IBM Corporation in the United States or other countries or both:

AIX/6000
AnyNet
AT
BookManager
C/370
CICS
CICS/ESA
DB2
DFSMS
DFSMSdfp
DFSMSdss
DFSMShsm
DFSMSrmm
DFSMS/MVS
DFSORT
ESCON
GDDM
Hiperbatch

IBM
IBMLink
IMS
Language Environment
Library Reader
Lotus Notes
MVS/ESA
MVS/SP
MVS/XA
OS/2
OS/390
RACF
RETAIN
RMF
SecureWay
SOMobjects
SystemView
System/390
S/390
TME 10
UNIX
VM/ESA
VTAM
Windows

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

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

Lotus and Lotus Notes are trademarks of Lotus Development Corporation in the United States, or other countries, or both.

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

# RACF Glossary

This glossary defines technical terms and abbreviations used in RACF documentation. If you do not find the term you are looking for, refer to the index of the appropriate RACF manual or view the IBM Glossary of Computing Terms, located at:
http://www.ibm.com/networking/nsg/nsgmain.htm

## Sequence of Entries

For purposes of clarity and consistency of style, this glossary arranges the entries alphabetically on a letter-by-letter basis, which means:

- Only the letters of the alphabet are used to determine sequence, and
- Special characters and spaces between words are ignored.

## Organization of Entries

Each entry consists of:

- A single-word term,
- A multiple-word term,
- An abbreviation for a term, or
- An acronym for a term.

This entry is followed by a commentary, which includes one or more items (definitions or references) and is organized as follows:

1. An item number, if the commentary contains two or more items.
2. A usage label, indicating the area of application of the term, for example, "In programming," or "In TCP/IP." Absence of a usage label implies that the term is generally applicable to IBM, or to data processing.
3. A descriptive phrase, stating the basic meaning of the term. The descriptive phrase is assumed to be preceded by "the term is defined as...". The part of speech being defined is indicated by the opening words of the descriptive phrase: "To ..." indicates a verb, and "Pertaining to ..." indicates a modifier. Any other wording indicates a noun or noun phrase.
4. Annotative sentences, providing additional or explanatory information.
5. References, pointing to other entries or items in the dictionary.

## References

The following cross-references are used in this glossary:

- **Contrast with:** This refers to a term that has an opposed or substantively different meaning.
- **See:** This refers the reader to (a) a related term, (b) a term that is the expanded form of an abbreviation or acronym, or (c) a synonym or more preferred term.
- **Synonym for:** This indicates that the term has the same meaning as a preferred term, which is defined in its proper place in the glossary.
- **Synonymous with:** This is a reference from a defined term to all other terms that have the same meaning.
- **Obsolete term for:** This indicates that the term should not be used and refers the reader to the preferred term.

## Selection of Terms

A term is the word or group of words being defined. In this glossary, the singular form of the noun and the infinitive form of the verb are the terms most often selected to be defined. If the term has an acronym or abbreviation, it is given in parentheses immediately following the term. The abbreviation's definition serves as a pointer to the term it abbreviates, and the acronym's definition serves as a pointer to the term it represents.

# A

**access.** The ability to use a protected resource.

**access authority.** (1) The privileges granted to a particular user or group when accessing a protected resource (such as the ability to read or to update a data set). For resources protected by RACF profiles, the access authorities are NONE, EXECUTE, READ, UPDATE, CONTROL, and ALTER. These authorities are hierarchical, with READ also granting EXECUTE, UPDATE granting READ, and so forth. (2) RACF also has access authorities of READ, WRITE, and EXECUTE (or SEARCH) when dealing with files and directories in the HFS. Note that these authorities are not hierarchical, and HFS files are not protected by RACF profiles, although they do have access authorities.

## Glossary

**access list.** Synonym for *standard access list*. Contrast with *conditional access list*.

**ACEE.** (accessor environment element) A control block that contains a description of the current user's security environment, including user ID, current connect group, user attributes, and group authorities. An ACEE is constructed during user identification and verification. See *ENVR object*.

**ADAU.** See *automatic direction of application updates*.

**ADSP.** See *automatic data set protection*.

**ADSP attribute.** A user attribute that establishes an environment in which all permanent DASD data sets created by the user are automatically defined to RACF and protected with a discrete profile. See *automatic data set protection*.

**Advanced Program-to-Program Communication (APPC).** A set of interprogram communication services that support cooperative transaction processing in an SNA network. APPC is the implementation, on a given system, of SNA's LU type 6.2. See *LU type 6.2* and *APPC/MVS*.

**APF-authorized.** A type of system authorization using the authorized program facility (APF) that allows an installation to identify system or user programs that can use sensitive system functions. To maintain system security and integrity, a program must be authorized by the APF before it can access restricted functions, such as supervisor calls (SVC) or SVC paths.

**API.** See *application programming interface*.

**APPC.** See *Advanced Program-to-Program Communication*.

**APPC application.** See *transaction program (TP)*.

**APPC/MVS.** The implementation of SNA's LU 6.2 and related communication services in the MVS base control program.

**application programming interface (API).** A software interface that enables applications to communicate with each other. An API is the set of programming language constructs or statements that can be coded in an application program to obtain the specific functions and services provided by an underlying operating system or service program.

**application user identity.** An alternate name by which a RACF user can be known to an application.

**appropriate privileges.** Describes which users can perform an action (such as execute a command, issue a syscall, and so forth) in a UNIX environment. Usually refers to having superuser authority or an appropriate subset of superuser authority.

**attribute.** See *user attribute* and *group-related user attribute*.

**AUDIT request.** The issuing of the RACROUTE macro with REQUEST=AUDIT specified. An AUDIT request is a general-purpose security request that a resource manager can use to audit.

**AUDITOR attribute.** A user attribute that allows the user to specify logging options on the RACF commands and list any profile (including its auditing options) using the RACF commands. Contrast with *group-AUDITOR attribute*.

**AUTH request.** The issuing of the RACROUTE macro with REQUEST=AUTH specified. The primary function of an AUTH request is to check a user's authorization to a RACF-protected resource or function. The AUTH request replaces the RACHECK function. See *authorization checking*.

**authentication.** (1) Verification of the identity of a user or the user's eligibility to access an object. (2) Verification that a message has not been altered or corrupted. (3) A process used to verify the user of an information system or protected resources. See also *password*.

**authority.** The right to access objects, resources, or functions. See *access authority, class authority,* and *group authority*.

**authorization checking.** The action of determining whether a user is permitted access to a protected resource. Authorization checking refers to the use of RACROUTE REQUEST=AUTH, RACROUTE REQUEST=FASTAUTH, or any of the RACF callable services unless otherwise stated. Note, however, that other RACF functions can also perform authorization checking as a part of their processing. For example, RACROUTE REQUEST=VERIFY can also check a user's authority to use a terminal or application.

**automatic command direction.** An RRSF function that enables RACF to automatically direct certain commands to one or more remote nodes after running the commands on the issuing node. Commands can be automatically directed based on who issued the command, the command name, or the profile class related to the command. Profiles in the RRSFDATA class control to which nodes commands are automatically directed. See *automatic direction of application updates, automatic password direction,* and *command direction.*

**automatic data set protection (ADSP).** A system function, enabled by the SETROPTS ADSP specification and the assignment of the ADSP attribute to a user with ADDUSER or ALTUSER, that causes all permanent data sets created by the user to be automatically defined to RACF with a discrete RACF profile.

**automatic direction.** See *automatic command direction*, *automatic password direction*, and *automatic direction of application updates*.

**automatic direction of application updates.** An RRSF function that automatically directs ICHEINTY and RACROUTE macros that update the RACF database to one or more remote systems. Profiles in the RRSFDATA class control which macros are automatically directed, and to which nodes. See *automatic command direction* and *automatic password direction*.

**automatic password direction.** An RRSF function that extends password synchronization and automatic command direction to cause RACF to automatically change the password for a user ID on one or more remote nodes after the password for that user ID is changed on the local node. Profiles in the RRSFDATA class control for which users and nodes passwords are automatically directed. See *password synchronization*, *automatic command direction*, and *automatic direction of application updates*.

**automatic profile.** A tape volume profile that RACF creates when a RACF-defined user protects a tape data set. When the last data set on the volume is deleted, RACF automatically deletes the tape volume profile. Contrast with *nonautomatic profile*.

# B

**backup data set.** A data set in the backup RACF database. For each data set in the primary RACF database, an installation should define a corresponding backup data set. See *backup RACF database*.

**backup RACF database.** A RACF database that reflects the contents of the primary RACF database. Backup RACF databases may be designated in the data set name table (ICHRDSNT) or specified at IPL time. You can switch to a backup database without a re-IPL if the primary RACF database fails. See *primary RACF database*.

**base segment.** The portion of a RACF profile that contains the fundamental information about a user, group, or resource. The base segment contains information that is common to all applications that use the profile.

**BER.** This term represents the Basic Encoding Rules specified in ISO 8825 for encoding data units described in abstract syntax notation 1 (ASN.1). The rules specify the encoding technique, not the abstract. See also *DER*.

**block update command (BLKUPD).** A RACF diagnostic command used to examine or modify the content of individual physical records in a RACF data set.

# C

**cache structure.** A coupling facility structure that contains data accessed by systems in a sysplex. For more information, see *OS/390 SecureWay Security Server RACF System Programmer's Guide*.

**callable service.** In OS/390 UNIX System Services, a request by an active process for a service. Synonymous with *syscall*.

**category.** See *security category*.

**CDMF.** See *Commercial Data Masking Facility*.

**CDT.** See *class descriptor table*.

**certificate.** See *digital certificate.*

**certificate authority.** An organization that issues digital certificates. The certificate authority authenticates the certificate owner's identity and the services that the owner is authorized to use, issues new certificates, renews existing certificates, and revokes certificates belonging to users who are no longer authorized to use them.

**certificate-authority certificate.** A type of certificate managed by RACF. See *digital certificate*.

**certificate name filter.** A general resource profile created by the RACDCERT MAP command that maps a digital certificate to multiple user IDs in order to simplify administration of certificates, conserve storage space in the RACF database, maintain accountability, or maintain access control granularity.

**CICS.** See *Customer Information Control System*.

**class.** A collection of RACF-defined entities (users, groups, and resources) with similar characteristics. Classes are defined in the class descriptor table (CDT), except for the USER, GROUP, and DATASET classes.

**class authority (CLAUTH).** An attribute enabling a user to define RACF profiles in a class defined in the class descriptor table. A user can have class authorities to zero or more classes.

**class descriptor table (CDT).** A table consisting of an entry for each class except the USER, GROUP, and DATASET classes. The CDT contains the classes supplied by IBM and the installation-defined classes.

**classification model 1.** See *single-subsystem scope*.

**classification model 2.** See *multiple-subsystem scope*.

**CLAUTH attribute.** See *class authority*.

**command direction.** An RRSF function that allows a user to issue a command from one user ID and direct that command to run in the RACF address space on the same system or on a different RRSF node, using the

same or a different user ID. Before a command can be directed from one user ID to another, a user ID association must be defined between them using the RACLINK command.

**command prefix facility (CPF).** An MVS facility that provides a registry for command prefixes. CPF ensures that two or more subsystems do not have the same or overlapping command prefixes for MVS operator commands.

**Commercial Data Masking Facility (CDMF).** An encryption function that uses a weaker key (40 bit) of the Data Encryption Standard (DES) algorithm. RACF uses CDMF to mask the data portion of RRSF transaction processing message packets. CDMF is part of the IBM Common Cryptographic Architecture.

**common programming interface (CPI).** An evolving application programming interface (API), supplying functions to meet the growing demands from different application environments and to achieve openness as an industry standard for communications programming. CPI-C provides access to interprogram services such as sending and receiving data, synchronizing processing between programs, and notifying a partner of errors in the communication.

**conditional access list.** The portion of a resource profile that specifies the users and groups that may access the resource at a specified level when a specified condition is true. For example, with program access to data sets, the condition is that the user must be executing the program specified in the access list. Contrast with *standard access list*.

**coordinator system.** In a RACF data sharing group, the system on which the system operator or administrator enters a RACF command that is propagated throughout the group. Contrast with *peer system*.

**coupling facility.** The hardware element that provides high-speed caching, list processing, and locking functions in a sysplex.

**CPF.** See *command prefix facility*.

**CPI-C.** See *common programming interface*.

**current connect group.** The group specified by a user when logging on to the system, or the user's default group if the user did not specify a group when logging on. With SETROPTS NOGRPLIST in effect, RACF uses the user's authority and this group's authority during access checking. With SETR GRPLIST in effect, RACF includes the authority of the user's other groups, if any, but the user still has only one "current connect group". You can use the &RACGPID variable in members of GLOBAL profiles to refer to the user's current connect group.

**current security label.** The security label that RACF uses in RACF authorization checking if the SECLABEL class is active. For interactive users, this is the security label specified when the user logged on, or (if no security label was specified) the default security label in the user's user profile. For batch jobs, this is the security label specified in the SECLABEL operand of the JOB statement, or (if no security label was specified) the user's current security label in the user profile associated with the job.

**Customer Information Control System (CICS).** A program licensed by IBM that provides online transaction processing services and management for critical business applications. CICS runs on many platforms (from the desktop to the mainframe) and is used in various types of networks that range in size from a few terminals to many thousands of terminals. The CICS application programming interface (API) enables programmers to port applications among the hardware and software platforms on which CICS is available. Each product in the CICS family can interface with the other products in the CICS family, thus enabling interproduct communication.

# D

**DASDVOL authority.** A preferred alternative to assigning the OPERATIONS or group-OPERATIONS attribute, DASDVOL authority allows you to authorize operations personnel to access only those volumes that they must maintain. Using DASDVOL authority is also more efficient for functions such as volume dumping, because only one authorization check for the volume needs to be issued, instead of individual requests for each data set on the volume. Note that modern data management software (such as DFSMSdss) does not require DASDVOL authority. Contrast with *OPERATIONS attribute*, and *group-OPERATIONS attribute*.

**Data Lookaside Facility (DLF).** A facility that processes DLF objects. A DLF object contains data from a single data set managed by Hiperbatch. The user (an application program) is connected to the DLF object, and the connected user can then access the data in the object through normal QSAM or VSAM macro instructions.

**data security.** The protection of data from intentional or unintentional unauthorized disclosure, modification, or destruction.

**data security monitor (DSMON).** A RACF auditing tool that produces reports enabling an installation to verify its basic system integrity and data security controls.

**data set profile.** A profile that provides RACF protection for one or more data sets. The information in the profile can include the data set profile name, profile

owner, universal access authority, access list, and other data. See *discrete profile* and *generic profile*.

**data sharing group, RACF.** A collection of one or more instances of RACF in a sysplex that have been identified to XCF and assigned to the group defined for RACF sysplex data sharing. RACF joins group IRRXCF00 when enabled for sysplex communication.

**data sharing mode.** An operational RACF mode that is available when RACF is enabled for sysplex communication. Data sharing mode requires installation of coupling facility hardware.

**DB2 administrative authority.** A set of privileges, often covering a related set of objects, and often including privileges that are not explicit, have no name, and cannot be specifically granted. For example, the ability to terminate any utility job is included in the SYSOPR authority.

**DB2 explicit privilege.** A privilege that has a name, and is held as the result of an SQL GRANT statement.

**DCE.** See *Distributed Computing Environment*.

**default group.** The group specified in a user profile that provides a default current connect group for the user. See *current connect group*.

**DEFINE request.** The issuing of the RACROUTE macro with REQUEST=DEFINE specified or using a RACF command to add or delete a resource profile causes a DEFINE request. The DEFINE request replaces the RACDEF function.

**delegation.** The act of giving users or groups the necessary authority to perform RACF operations.

**DER.** This term represents the Distinguished Encoding Rules, which are a subset of the Basic Encoding Rules. See also *BER*.

**digital certificate.** A digital document that binds a public key to the identity of the certificate owner, thereby enabling the certificate owner to be authenticated. A certificate is issued by a certificate authority.

RACF can manage three types of digital certificates:

- **Certificate-authority certificate.** A certificate associated with a certificate authority and is used to verify signatures in other certificates.
- **Site certificate.** A certificate associated with a server, or network entity other than a user or certificate authority.
- **User certificate.** A certificate associated with a RACF user ID that is used to authenticate the user's identity.

**DIRAUTH request.** The issuing of the RACROUTE macro with REQUEST=DIRAUTH specified. A DIRAUTH request works on behalf of the message-transmission managers to ensure that the receiver of a message meets security-label authorization requirements.

**directed command.** A RACF command that is issued from a user ID on an RRSF node. It runs in the RACF subsystem address space on the same or a different RRSF node under the authority of the same or a different user ID. A directed command is one that specifies AT or ONLYAT. See *command direction* and *automatic command direction*.

**discrete profile.** A resource profile that provides RACF protection for a single resource. Contrast with *generic profile* and *fully-qualified generic profile*.

**discretionary access control.** An access control environment in which the resource owner determines who can access the resource. Contrast with *mandatory access control*.

**disjoint.** Pertaining to security labels, when the set of security categories that defines the first does not include the set of security categories that defines the second, and the set of security categories that defines the second does not include the set of security categories that defines the first. This also means that the first does not dominate the second and the second does not dominate the first. See *dominate*.

**Distributed Computing Environment (DCE).** The Open Group specification (or a product derived from this specification) that assists in networking. DCE provides such functions as authentication, directory service (DS), and remote procedure call (RPC).

**DLF object.** When DLF is active, the first attempt to access a QSAM or VSAM data set defined to DLF creates a DLF object. A DLF object contains data from a single data set managed by Hiperbatch. The user (an application program) is connected to the DLF object, and the connected user can then access the data in the object through normal QSAM or VSAM macro instructions.

**dominate.** One security label dominates a second security label when the security level that defines the first is equal to or greater than the security level that defines the second, and the set of security categories that defines the first includes the set of security categories that defines the second.

**DSMON.** See *data security monitor*.

# E

**effective group identifier (effective GID).** When the user connects to the system (for example, logs on to a TSO/E session), one group is selected as the user's current group. When a user becomes an OS/390 UNIX user, the GID of the user's current group becomes the effective GID of the user's process. The user can access

resources available to members of the user's effective GID. See *OS/390 UNIX group identifier (GID)* and contrast with *real GID*.

**effective user identifier (effective UID).** When a user becomes an OS/390 UNIX user, the UID from the user's RACF user profile becomes the effective UID of the user's process. The system uses the effective UID to determine if the user is a file owner. See *OS/390 UNIX user identifier (UID)* and contrast with *real UID*.

**ENVR object.** A transportable form of the ACEE that can be used within a single system to create the original ACEE without accessing the RACF database. It can be used, with limits, elsewhere in a single sysplex to recreate the original ACEE without accessing the RACF database.

**entity.** A user, group, or resource (for example, a DASD data set) that is defined to RACF.

**erase-on-scratch.** The physical overwriting of data on a DASD data set when the data set is deleted (scratched).

**EXTRACT request.** The issuing of the RACROUTE macro with REQUEST=EXTRACT specified. An EXTRACT request retrieves or replaces certain specified fields from a RACF profile or encodes certain clear-text (readable) data. The EXTRACT request replaces the RACXTRT function.

# F

**failsoft processing.** (1) Processing that occurs when no data sets in the primary RACF database are available (RACF is installed but inactive). RACF cannot make decisions to grant or deny access. The operator is prompted frequently to grant or deny access to data sets. The resource manager decides on the action for general resource classes with a return code of 4. (2) Failsoft processing can also occur as the result of RVARY INACTIVE (temporary failsoft) or as the result of a serious system error requiring a re-IPL (permanent failsoft).

**FASTAUTH request.** The issuing of the RACROUTE macro with REQUEST=FASTAUTH specified. The primary function of a FASTAUTH request is to check a user's authorization to a RACF-protected resource or function. A FASTAUTH request uses only in-storage profiles (brought into storage using RACF functions such as RACROUTE REQUEST=LIST) for faster performance than an AUTH request. The FASTAUTH request replaces the FRACHECK function. See *authorization checking*.

**field-level access checking.** The RACF facility by which a security administrator can control access to segments, other than the base segment, in a RACF profile and fields in those segments.

**file permission bits.** In OS/390 UNIX System Services, information about a file that is used, along with other information, to determine if a process has read, write, or execute/search permission to a file or directory. The bits are divided into three parts, which are owner, group, and other.

**file security packet (FSP).** In OS/390 UNIX System Services, a control block containing the security data (file's owner OS/390 UNIX user identifier (UID), owner OS/390 UNIX group identifier (GID), and the permission bits) associated with the file. This data is stored with the file in the OS/390 UNIX file system.

**FMID.** See *function modification identifier*.

**FSP.** See *file security packet*.

**file transfer program (FTP).** In the Internet suite of TCP/IP-related protocols, an application-layer protocol that transfers bulk-data files between machines or hosts.

**FRACHECK request.** RACROUTE REQUEST=FASTAUTH replaces the FRACHECK function. See *FASTAUTH request*.

**FTP.** See *File Transfer Protocol*.

**full-qualified generic profile.** A DATASET profile that was defined using the GENERIC operand and has a name that contains no generic characters. A fully-qualified generic profile protects only resources whose names exactly match the name of the profile. Contrast with *discrete profile* and *generic profile*.

**function modification identifier (FMID).** A 7-character identifier that is used in elements associated with OS/390 to identify the release of the element.

# G

**GDG.** See *generation data group*.

**general resource.** Any resource, other than an MVS data set, that is defined in the class descriptor table (CDT). General resources include DASD volumes, tape volumes, load modules, terminals, IMS and CICS transactions, and installation-defined resource classes.

**general resource profile.** A profile that provides RACF protection for one or more general resources. The information in the profile can include the general resource profile name, profile owner, universal access authority, access list, and other data.

**general user.** A user who has limited RACF privileges, such as logging on, accessing resources, and creating data sets. General users typically use and create RACF-protected resources, but have no authority to administer resources other than their own.

**generation data group (GDG).** A collection of data sets with the same base name, such as PAYROLL, that are kept in chronological order. Each data set in the GDG is called a generation data set, and has a name such as PAYROLL.G0001V00, PAYROLL.G0002V00, and so forth.

**generic profile.** A resource profile that can provide RACF protection for zero or more resources. The resources protected by a generic profile have similar names and identical security requirements, though with RACFVARS, a generic profile can protect resources with dissimilar names, too. For example, a generic data set profile can protect one or more data sets. Contrast with *discrete profile*.

**GID.** See *OS/390 UNIX group identifier (GID)*.

**global access checking.** The ability to allow an installation to establish an in-storage table of default values for authorization levels for selected resources. RACF refers to this table before performing normal RACROUTE REQUEST=AUTH processing and grants the request without performing an AUTH request if the requested access authority does not exceed the global value. RACF uses this table to process AUTH requests faster and with less overhead (no checking of access lists, no auditing) when you have resources for which you decide to grant access to all users, except those with restricted user IDs. If the requested access does not exceed the access granted by the table, RACF bypasses most of its normal AUTH processing. Global access checking can grant the user access to the resource, but it cannot deny access.

**global resource serialization.** An OS/390 mechanism using ENQ with the SYSTEMS option (or, in some older programs, the RESERVE option) to serialize resources across multiple OS/390 images. It is used by RACF to serialize access to its database and to in-storage tables and buffers.

**globally RACLISTed profiles.** In-storage profiles for RACF-defined resources that are created by RACROUTE REQUEST=LIST and that are anchored from an ACEE. Globally RACLISTed in-storage profiles are shared across a system, such as the way that in-storage profiles created by SETROPTS RACLIST are shared. Contrast with *locally RACLISTed profiles*.

**group.** A collection of RACF-defined users who can share access authorities for protected resources.

**group-ADSP attribute.** A group-related user attribute similar to the ADSP attribute for a user, but assigned by using the CONNECT command to restrict its effect to those cases where the user creates data sets with that group as the high level qualifier of the data set name (or as determined by the naming convention table or exit).

**group-AUDITOR attribute.** A group-related user attribute similar to the AUDITOR attribute for a user, but assigned by using the CONNECT command to restrict the user's authority to resources that are within the scope of the group. Contrast with *AUDITOR attribute*.

**group authority.** An authority specifying which functions a user can perform in a group. The group authorities are USE, CREATE, CONNECT, and JOIN.

**group data set.** A RACF-protected data set in which either the high-level qualifier of the data set name or the qualifier supplied by an installation-naming convention table or exit routine is a RACF group name.

**group-GRPACC attribute.** A group-related user attribute similar to the GRPACC attribute for a user, but assigned by using the CONNECT command to restrict its effect to the specific group. Contrast with *GRPACC attribute*.

**group ID.** Obsolete term for *group name*.

**group name.** A string of 1–8 characters that identifies a group to RACF. The first character must be A through Z, # (X'7B'), $ (X'5B'), or @ (X'7C'). The rest can be A through Z, #, $, @, or 0 through 9.

**group-OPERATIONS attribute.** (1) A group-related user attribute similar to the OPERATIONS attribute for a user, but assigned by using the CONNECT command to restrict its effect to those resources that are within the scope of the group. (2) If a person needs to perform maintenance activities on DASD volumes, it is more efficient (for RACF processing) and better (for limiting the resources the person can access) to give the person authority to those volumes using the PERMIT command than to assign the person the OPERATIONS or group-OPERATIONS attribute. Contrast with *DASDVOL authority* and *OPERATIONS attribute*.

**group profile.** A profile that defines a group. The information in the profile includes the group name, profile owner, and users in the group.

**grouping profile.** A profile in a resource group class.

**group-REVOKE attribute.** Assigned through the CONNECT command that prevents the user from using that group as the current connect group. Also prevents RACF from considering that group during authorization checking.

**group-SPECIAL attribute.** A group-related user attribute similar to the SPECIAL user attribute, but it is assigned by the CONNECT command to restrict the user's authority to users, groups, and resources within the scope of the group. Within this scope, it gives the user full control over everything except auditing options. However, it does not give the user authority to change global RACF options that will affect processing outside the group's scope. Contrast with *SPECIAL attribute*.

**group-related user attribute.** A user attribute, assigned at the group level, that enables the user to control the resource, group, and user profiles associated with the group and its subgroups. Group-related user attributes include group-SPECIAL attribute, group-AUDITOR attribute, and group-OPERATIONS attribute. Contrast with *user attribute*.

**GRPACC attribute.** With this attribute, any group data sets that the user defines to RACF (through the ADSP attribute, the PROTECT operand on the DD statement, or the ADDSD command) are automatically made accessible to other users in the group at the UPDATE level of access authority if the user defining the profile is a member of the group. Contrast with *group-GRPACC attribute*.

# H

**HFS.** See *hierarchical file system*.

**hierarchical file system (HFS).** The file system for OS/390 UNIX System Services that organizes data in a tree-like structure of directories.

# I

**ICB.** See *inventory control block*.

**interprocess communication facilities (IPC).** IPC facilities are services that allow different processes to communicate. Message passing (using message queues), semaphore sets, and shared memory services are forms of interprocess communication facilities.

**inventory control block (ICB).** The first block in a RACF database. The ICB contains a general description of the database and, for the master primary data set, holds the RACF global options specified by SETROPTS.

**ICHRIN03.** See *started procedures table*.

**IPC.** See *interprocess communication facilities*.

**issuer's distinguished name (IDN).** The X.509 name that is associated with a certificate authority.

# K

**kernel.** The part of OS/390 UNIX System Services that provides support for such services as UNIX I/O, process management, and general UNIX functionality.

**kernel address space.** The address space in which the OS/390 UNIX System Services kernel runs. See *kernel*.

**key.** In cryptography, a sequence of symbols that is used with a cryptographic algorithm for encrypting or decrypting data. See *private key* and *public key*.

**key ring.** A named collection of certificates for a specific user or server application used to determine the trustworthiness of a client or peer entity.

# L

**link pack area (LPA).** An area of virtual storage containing reenterable routines from system libraries that are loaded at IPL time and can be used concurrently by all tasks in the system. The LPA presence in main storage saves loading time.

**LIST request.** The issuing of the RACROUTE macro with REQUEST=LIST specified. A LIST request builds in-storage profiles for a RACF general resource class. The LIST request replaces the RACLIST function.

**list-of-groups checking.** A RACF option (SETROPTS GRPLIST) that enables a user to access all resources available to all groups of which the user is a nonrevoked member, regardless of the user's current connect group. For any particular resource, RACF allows access based on the highest access among the groups in which the user is a member.

**local logical unit (local LU).** A logical unit that resides on the local system. Contrast with *partner logical unit (partner LU)*, or *remote logical unit (remote LU)*, which typically resides on a remote system. When both the local and partner LUs reside on the same system, the LU through which communication is initiated is the local LU, and the LU through which communication is received is the partner LU.

**local mode.** An RRSF node is operating in local mode when it has no RRSF logical node connection with any other RRSF node.

**local transaction program (local TP).** A transaction program that resides on the local system. Contrast with *partner transaction program (partner TP)*, which typically resides on a remote system.

**logging.** The recording of audit data about specific events.

**logical connection.** See *RRSF logical node connection*.

**logical unit (LU).** A type of network accessible unit that enables users to gain access to network resources and communicate with each other.

**locally RACLISTed profiles.** In-storage profiles for RACF-defined resources that are created by RACROUTE REQUEST=LIST and that are anchored from an ACEE. Locally RACLISTed in-storage profiles are not shared across a system, the way that in-storage profiles created by SETROPTS RACLIST are shared. Contrast with *globally RACLISTed profiles*.

**LPA.** See *link pack area*.

**LU.** See *logical unit*.

# M

**MAC.** See *mandatory access control*.

**main system.** The system on a multisystem RRSF node that is designated to receive most of the RRSF communications sent to the node.

**managed user ID association.** A user ID association in which one of the associated user IDs is a managing user ID, and the other is a managed user ID. The managing user ID can run allowed RACF commands under the authority of the managed user ID. The managed user ID cannot run commands under the authority of the managing user ID. A managed user ID association does not allow password synchronization between the associated user IDs. Contrast with *peer user ID association*.

**mandatory access control (MAC).** A means of restricting access to objects on the basis of the sensitivity (as represented by a label) of the information contained in the objects and the formal authorization (clearance) of subjects to access information of such sensitivity.

**mask.** A technique to provide protection against casual viewing of a password that has been defined or altered, when an encryption function is not available.

**master primary data set.** The first data set activated in the primary RACF database.

**MCS.** See *multiple console support*.

**MCS console.** A non-SNA device defined to MVS that is locally attached to an MVS system and is used to enter commands and receive messages.

**member.** A user belonging to a group.

**member profile.** A profile that defines a member and security level for that member.

**member system.** Any one of the MVS system images in a multisystem RRSF node.

**modeling.** See *profile modeling*.

**multiple console support (MCS).** The operator interface in an MVS system.

**multiple-subsystem scope.** A RACF classification model used in conjunction with the RACF/DB2 external security module to construct DB2 resource names. Default for the highest-level qualifier is the DB2 subsystem or group name.

**multisystem node.** See *multisystem RRSF node*.

**multisystem RRSF node.** An RRSF node consisting of multiple MVS system images that share the same RACF database. One of the systems is designated to be the main system, and it receives the unsolicited RRSF communications sent to the node.

**multi-subsystem scope.** A classification model used in conjunction with the RACF/DB2 external security module to construct DB2 classes with the subsystem ID as part of the class name. Contrast with *single-subsystem scope*.

**MVS.** (multiple virtual storage) The mainframe operating system that allows multiple users to work simultaneously using the full amount of virtual storage.

# N

**NCSC.** National Computer Security Center. The part of the U.S. Department of Defense that determines defense and security criteria.

**network-qualified name.** An identifier for a partner LU in the form *netid.luname*, where *netid* is a 1–8 character network identifier and *luname* is a 1–8 character LU name.

**node.** See *RRSF node*.

**non-data sharing mode.** One of two normal modes of operation when RACF is enabled for sysplex communication and is the mode in which RACF communicates information using sysplex facilities to other instances of RACF, but does not make use of the coupling facility in doing so.

**nonautomatic profile.** A tape volume profile that RACF creates when an RDEFINE command is issued or when tape data set protection is not active. A tape volume profile created in this manner is called a nonautomatic profile because RACF never deletes the profile except in response to the RDELETE command. Contrast with *automatic profile*.

# O

**operator identification card (OIDCARD).** A small card with a magnetic stripe encoded with unique characters and used to verify the identity of a terminal operator to RACF on an OS/390 system.

**OPERATIONS attribute.** A user attribute that grants the equivalent of ALTER access to all data sets unless the user or one of the user's connect groups appears explicitly in the access list of a data set's profile. If a user needs to perform maintenance activities on DASD volumes, granting DASDVOL authority to those volumes using the PERMIT command is preferred over assigning the OPERATIONS or group-OPERATIONS attribute. Note that most modern DASD maintenance programs do not require the OPERATIONS attribute. Contrast with *DASDVOL authority* and *group-OPERATIONS attribute*.

**OS/390.** A program licensed by IBM that not only includes and integrates functions previously provided by many IBM software products, including the MVS operating system, but also:

1. Is an open, secure operating system for the IBM S/390 family of enterprise servers
2. Complies with industry standards
3. Is Year 2000 ready and enabled for network computing and e-business
4. Supports technology advances in networking server capability, parallel processing, and object-oriented programming

**OS/390 UNIX group identifier (GID).** A number between 0 and 2 147 483 647 that identifies a group of users to OS/390 UNIX. The GID is associated with a RACF group name when it is specified in the OMVS segment of the group profile. See *real GID*. Contrast with *effective group identifier (effective GID)*.

**OS/390 UNIX System Services (OS/390 UNIX).** The set of functions provided by the shells, utilities, kernel, file system, debugger, Language Environment, and other elements of the OS/390 operating system that allows users to write and run application programs that conform to UNIX standards.

**OS/390 UNIX user identifier (UID).** A number between 0 and 2 147 483 647 that identifies a user to OS/390 UNIX. The UID is associated with a RACF user ID when it is specified in the OMVS segment of the user profile. It can be contained in an object of type uid_t, that is used to identify a system user. When the identity of the user is associated with a process, a UID value is referred to as a real UID, an effective UID, or an (optional) saved set UID. See *real UID*. Contrast with *effective user identifier (effective UID)*.

**owner.** The user or group that creates a profile, or is specified as the owner of a profile. The owner can modify, list, or delete the profile.

# P

**PADS.** See *program access to data sets (PADS)*.

**partner logical unit (partner LU).** A logical unit that typically resides on a remote system. Often synonymous with *remote logical unit (remote LU)*. Contrast with *local logical unit (local LU)*, which resides on the local system. When both the local and partner LUs reside on the same system, the LU through which communication is initiated is the local LU, and the LU through which communication is received is the partner LU.

**partner transaction program (partner TP).** A transaction program that resides on a remote system. Contrast with *local transaction program (local TP)*, which typically resides on the local system.

**PassTicket.** An alternative to the RACF password that permits workstations and client machines to communicate with the host. It allows a user to gain access to the host system without sending the RACF password across the network.

**password.** A string of characters known to a user who must specify it to gain full or limited access to a system and to the data stored within it. RACF uses a password to verify the identity of the user.

**password synchronization.** An option that can be specified when a peer user ID association is defined between two user IDs. If password synchronization is specified for a user ID association, then whenever the password for one of the associated user IDs is changed, the password for the other user ID is automatically changed to the newly defined password. See *automatic password direction*.

**peer system.** In a RACF data sharing group, any system to which RACF propagates a command entered by the system operator or administrator. Contrast with *coordinator system*.

**peer user ID association.** A user ID association that allows either user ID to run allowed RACF commands under the authority of the other user ID using command direction. A peer user ID association can also establish password synchronization between the associated user IDs. Contrast with *managed user ID association*.

**permission bits.** In OS/390 UNIX System Services, part of security controls for directories and files stored in the hierarchical file system (HFS). Used to grant read, write, search (just directories), or execute (just files) access to owner, file or directory owning group, or all others.

**persistent verification (PV).** A VTAM security option for conversation-level security between two logical units (LUs) that provides a way of reducing the number of password transmissions by eliminating the need to provide a user ID and password on each attach (allocate) during multiple conversations between a user and a partner LU. The user is verified during the signon process and remains verified until the user has been signed off the partner LU.

**POSIX.** (Portable Operating System Interface For Computer Environments) An IEEE standard for computer operating systems.

**POSIT.** A number specified for each class in the class descriptor table that identifies a set of flags that control RACF processing options. See the description of the POSIT keyword in the *OS/390 SecureWay Security Server RACF Macros and Interfaces*.

**primary data set.** A data set in the primary RACF database. See *master primary data set*.

**primary RACF database.** The RACF database designated in the data set name table (ICHRDSNT), or specified at IPL time, that contains the RACF profiles used for authorization checking. The primary RACF database may consist of as many as 90 data sets. See *backup RACF database*.

**private key.** In public key cryptography, a key that is known only to its owner. Contrast with *public key*.

**problem state.** A state during which a processing unit cannot execute input/output and other privileged instructions. Contrast with *supervisor state*.

**process.** In OS/390 UNIX, a function created by a **fork()** request. See *task*.

**profile.** Data that describes the significant characteristics of a user, a group of users, or one or more computer resources. A profile contains a base segment, and optionally, a number of other segments. See *data set profile, discrete profile, general resource profile, generic profile, group profile,* and *user profile*.

**profile list.** A list of profiles indexed by class (for general resources) or by the high-level qualifier (for data set profiles) and built in storage by the RACF routines.

**profile modeling.** The ability for a user or an installation to copy information (such as universal access authority or access lists) from an existing resource profile when defining a new resource profile. This might occur automatically when using ADDSD based on the MODEL specification in a USER or group PROFILE, or manually with the FROM keyword of the ADDSD and RDEFINE commands, or with keywords on RACROUTE REQUEST=DEFINE.

**program access to data sets (PADS).** A RACF function that enables an authorized user or group of users to access one or more data sets at a specified access authority only while running a specified RACF-controlled program. See *program control*.

**program control.** A RACF function that enables an installation to control who can run RACF-controlled programs. See *program access to data sets*.

**protected resource.** A resource defined to RACF for the purpose of controlling access to the resource. Some of the resources that can be protected by RACF are DASD volumes, tape volumes, load modules, terminals, IMS and CICS transactions, and installation-defined resource classes.

**protected user ID.** A user ID that cannot enter the system by any means that requires a password, and cannot be revoked by invalid password attempts. Assigning a protected user ID to OS/390 UNIX, a UNIX daemon, or another important started task or subsystem assures that the ID cannot be used for other purposes, and that functions will not fail because the ID has been revoked.

**public key.** In public key cryptography, a key that is made available to everyone. Contrast with *private key*.

**public key cryptography.** Cryptography in which public keys and private keys are used for encryption and decryption. One party uses a common public key and the other party uses secret private key. The keys are complementary in that if one is used to encrypt data, the other can be used to decrypt it.

**PV.** See *persistent verification*.

# R

**RACDEF request.** The DEFINE function replaces the RACDEF function. See *DEFINE request*.

**RACF.** See *Resource Access Control Facility*.

**RACF/DB2 external security module.** A RACF exit point that receives control from the DB2 access control authorization exit point (DSNX@XAC) to handle DB2 authorization checks.

**RACF database.** The repository for the security information that RACF maintains.

**RACF data set.** One of the data sets comprising the RACF database.

**RACF-indicated.** Pertaining to a data set for which the RACF indicator is set on. If a data set is RACF-indicated, a user can access the data set only if a RACF profile or an entry in the global access checking table exists for that data set. On a system without RACF, a user cannot access a RACF-indicated data set until the indicator is turned off. For VSAM data sets, the indicator is in the catalog entry. For non-VSAM data sets, the indicator is in the data set control block (DSCB). For data sets on tape, the indicator is in the RACF tape volume profile of the volume that contains the data set.

**RACF manager.** The routines within RACF that provide access to the RACF database. Contrast with *RACF storage manager*.

**RACF-protected.** Pertaining to a resource that has either a discrete profile or an applicable generic profile. A data set that is RACF-protected by a discrete profile must also be RACF-indicated.

**RACF remote sharing facility (RRSF).** RACF services that function within the RACF subsystem address space to provide network capabilities to RACF.

**RACF remove ID utility.** A RACF utility that identifies references to user IDs and group names in the RACF database. The utility can be used to find

references to residual user IDs and group names or specified user IDs and group names. The output from this utility is a set of RACF commands that can be used to remove the references from the RACF database after review and possible modification. See *residual user ID*.

**RACF report writer.** A RACF function that produces reports on system use and resource use from information found in the RACF SMF records. However, the preferred method for producing RACF SMF reports is the RACF SMF unload utility (IRRADU00).

**RACF segment.** Obsolete term for *base segment*.

**RACF SMF data unload utility (IRRADU00).** A RACF utility that enables installations to create a sequential file from the security-relevant audit data. The sequential file can be viewed directly, used as input for installation-written programs, and manipulated with sort/merge utilities. It can also be uploaded to a database manager (such as DB2) to process complex inquiries and create installation-tailored reports. See *SMF records*.

**RACF storage manager.** Manages the allocation of storage for the RACF programs running on a system.

**RACHECK request.** The AUTH request replaces the RACHECK function. See *AUTH request*.

**RACINIT request.** The RACINIT request replaces the RACINIT function. See *VERIFY request*.

**RACLIST request.** The LIST request replaces the RACLIST function. See *LIST request*.

**RACLISTed profiles.** See *locally RACLISTed profiles* and *globally RACLISTed profiles*.

**RACROUTE macro.** An assembler macro that provides a means of calling RACF to provide security functions, including the *AUDIT request*, *AUTH request*, *DEFINE request*, *DIRAUTH request*, *EXTRACT request*, *FASTAUTH request*, *LIST request*, *SIGNON request*, *STAT request*, *TOKENBLD request*, *TOKENMAP request*, *TOKENXTR request*, *VERIFY request*, and *VERIFYX request*.

**RACSTAT request.** The STAT request replaces the RACSTAT function. See *STAT request*.

**RACXTRT request.** The EXTRACT request replaces the RACXTRT function. See *EXTRACT request*.

**RBA.** See *relative byte address*.

**read-only mode.** A recovery mode of operation when RACF is enabled for sysplex communication. Read-only mode does not allow updates to be made to the RACF database except for statistics generated during logon and job initiation.

**real GID.** The attribute of a process that, at the time of process creation, identifies the group of the user who created the process. See *OS/390 UNIX group identifier (GID)*. Contrast with *effective group identifier (effective GID)*.

**real UID.** The attribute of a process that, at the time of process creation, identifies the user who created the process. See *OS/390 UNIX user identifier (UID)*. Contrast with *effective user identifier (effective UID)*.

**relative byte address (RBA).** The address in the RACF database.

**remote logical unit (remote LU).** A logical unit that resides on a remote system. Often synonymous with *partner logical unit (partner LU)*. Contrast with *local logical unit (local LU)*, which typically resides on the local system.

**residual authority.** References in the RACF database to group names and user IDs that have been deleted.

**residual group name.** References in the RACF database to a group name that has been deleted.

**residual user ID.** References in the RACF database to a user ID that has been deleted.

**Resource Access Control Facility (RACF).** A program (licensed by IBM) that provides access control by identifying and verifying the users to the system, authorizing access to protected resources, logging detected unauthorized attempts to enter the system, logging unauthorized attempts to enter the system, and logging detected accesses to protected resources. RACF is included in the SecureWay Security Server for OS/390 and is also available as a separate program for the MVS and VM environments.

**resource grouping class.** A RACF class in which resource group profiles can be defined. A resource grouping class is related to another class, sometimes called a *member class*. For example, the resource grouping class GTERMINL is related to the class TERMINAL. See *resource group profile*.

**resource group profile.** A general resource profile in a resource grouping class. A resource group profile can provide RACF protection for one or more resources with *unlike* names. See *resource grouping class*.

**resource profile.** A profile that provides RACF protection for one or more resources. USER, GROUP, and CONNECT profiles are not resource profiles. The information in a resource profile can include the profile name, profile owner, universal access authority, access list, and other data. Resource profiles can be discrete profiles or generic profiles. See *discrete profile* and *generic profile*.

**RESTRICTED attribute.** A user attribute that can be assigned to a shared user ID, such as PUBLIC or ANONYMOS, or a user ID used with a certificate name filter, to prevent the user ID from being used to access

protected resources it is not specifically authorized to access. Restricted users cannot gain access to protected resources through global access checking, UACC, or an `ID(*)` entry on the access list.

**REVOKE attribute.** A user attribute that prevents a RACF-defined user from entering the system.

**role.** In Tivoli products, a functional grouping of user authorizations. A ROLE profile represents a role and identifies the authorizations associated with that role.

**RRSF.** See *RACF remote sharing facility*.

**RRSF logical node connection.** Two RRSF nodes are logically connected when they are properly configured to communicate through APPC/MVS, and they have each been configured by the TARGET command to have an OPERATIVE connection to the other.

**RRSF network.** Two or more RRSF nodes that have established RRSF logical node connections to each other.

**RRSF node.** An MVS system image or a group of MVS system images sharing a RACF database, which has been defined as an RRSF node, single-system RRSF node, or multisystem RRSF node to RACF by a TARGET command. See *RRSF logical node connection*.

**RTOKEN.** The RACF resource security token. An RTOKEN is an encapsulation or representation of the security characteristics of a resource. Resource managers, for example JES, can assign RTOKENs to the resources they manage; for example, JES spool files. See *UTOKEN* and *STOKEN*.

# S

**SAF.** See *System Authorization Facility*.

**secured signon.** A RACF function providing an alternative to the RACF password and also providing enhanced security across a network.

**SecureWay Security Server.** A licensed feature of OS/390 that is comprised of Resource Access Control Facility (RACF), DCE Security Server, Lightweight Directory Access Protocol (LDAP) Server, OS/390 Firewall Technologies, Open Cryptographic Enhanced Plug-ins (OCEP), and Network Authentication and Privacy Service.

**security.** See *data security*.

**security category.** An installation-defined name corresponding to a department or area within an organization whose members have similar security requirements.

**security classification.** The use of security categories, a security level, or both, to impose additional access controls on sensitive resources. An alternative way to provide security classifications is to use security labels.

**security label.** An installation-defined name that corresponds to a specific RACF security level with a set of zero or more security categories. This is equivalent to the NCSC term *sensitivity label*.

**security level.** An installation-defined name that corresponds to a numerical security level; the higher the number, the higher the security level.

**Security Server.** See *SecureWay Security Server*.

**security token.** A collection of identifying and security information that represents data to be accessed, a user, or a job. This contains a user ID, group name, security label, node of origin, and other information.

**segment.** A portion of a profile. The format of each segment is defined by a template.

**SETROPTS RACLISTed profiles.** See *globally RACLISTed profiles*.

**SFS.** See *Shared File System*.

**shared file system (SFS).** On VM/ESA, a part of CMS that lets users organize their files into groups known as directories and selectively share those files and directories with other users.

**signed-on-from list.** A list of user entries identifying those users who have been signed on from a partner LU to a local LU and is associated with persistent verification.

**SIGNON request.** The issuing of the RACROUTE macro with REQUEST=SIGNON specified. A SIGNON request is used to manage the signed-on-from lists associated with persistent verification.

**single-subsystem scope.** A classification model used in conjunction with the RACF/DB2 external security module to construct DB2 classes with the subsystem ID as part of the class name. Contrast with *multi-subsystem scope*.

**single-system node.** See *single-system RRSF node*.

**single-system RRSF node.** An RRSF node consisting of one MVS system image.

**site certificate.** A type of certificate managed by RACF. See *digital certificate*.

**SMF.** See *System Management Facilities*.

**SMF records.** (1) Records and system or job-related information collected by the System Management Facilities (SMF) and used in billing users, reporting reliability, analyzing the configuration, scheduling jobs, summarizing direct access volume activity, evaluating data set activity, profiling system resource use, and

maintaining system security. (2) Variable-length process or status records from the SMF data set that are written to the SMF log data set. These records vary in layout based on the type of system information they contain. See *RACF SMF unload utility*.

**SMS.** See *Storage Management Subsystem*.

**SNA.** See *System Network Architecture (SNA)*.

**source user ID.** The source half of a source user ID and target user ID pair that has an established user ID association between them. For command direction the source user ID is the user ID that issued the command that is being directed. For password synchronization the source user ID is the user ID whose password changed, causing a change to the password of the target user ID. Contrast with *target user ID*.

**SPECIAL attribute.** A user attribute that gives the user full control over all of the RACF profiles in the RACF database and allows the user to issue all RACF commands, except for commands and operands related to auditing. Contrast with *group-SPECIAL attribute*.

**split database.** A RACF database that has been divided among multiple data sets.

**standard access list.** The portion of a resource profile that specifies the users and groups that may access the resource and the level of access granted to each. Synonymous with *access list*. Contrast with *conditional access list*.

**started procedures table (ICHRIN03).** Associates the names of started procedures with specific RACF user IDs and group names. It can also contain a generic entry that assigns a user ID or group name to any started task that does not have a matching entry in the table. However, it is recommended that you use the STARTED class for most cases rather than the started procedures table.

**STAT request.** The issuing of the RACROUTE macro with REQUEST=STAT specified. A STAT request determines if RACF is active and (optionally) if a given resource class is defined to RACF and active. The STAT request replaces the RACSTAT function.

**STOKEN.** A UTOKEN associated with a user who has submitted work. See *UTOKEN* and *RTOKEN*.

**Storage Management Subsystem (SMS).** A component of MVS/DFP that is used to automate and centralize the management of storage by providing the storage administrator with control over data class, storage class, management class, storage group, and automatic class selection routine definitions.

**structure.** See *cache structure*.

**stub.** (1) A function that connects with the specified library, but remains outside the specified library. (2) A protocol extension procedure.

**subject's distinguished name (SDN).** The X.509 name in a digital certificate that is associated with the name of the subject.

**superuser.** In OS/390 UNIX System Services, a system user who operates with the special privileges needed to perform a specified administrative task.

**superuser authority.** In an OS/390 UNIX System Services operating system, the unrestricted authority to access and modify any part of the operating system, usually associated with the user who manages the system.

**supervisor.** The part of a control program that coordinates the use of resources and maintains the flow of processing unit operations. Synonym for *supervisory routine*.

**supervisor state.** A state during which a processing unit can execute input/output and other privileged instructions. Contrast with *problem state*.

**supervisory routine.** A routine, usually part of an operating system, that controls the execution of other routines and regulates the flow of work in a data processing system. Synonymous with *supervisor*.

**syscall.** See *callable service*.

**sysplex (system complex).** Multiple systems communicating and cooperating with each other through multisystem hardware elements and software services to process the installation's workloads.

**sysplex communication.** An optional RACF function that allows the system to use XCF services and communicate with other systems that are also enabled for sysplex communication.

**system complex.** See *sysplex*.

**system authorization facility (SAF).** An interface defined by MVS that enables programs to use system authorization services in order to control access to resources, such as data sets and MVS commands. SAF either processes security authorization requests directly or works with RACF, or other security product, to process them.

**system call.** In OS/390 UNIX System Services, a synonym for *callable service*.

**System Management Facility (SMF).** The part of the OS/390 operating system that collects and records system and job-related information used in billing users, reporting reliability, analyzing the configuration, scheduling jobs, summarizing direct access volume activity, evaluating data set activity, profiling system

resource use, and maintaining system security. The information is recorded in the SMF log data set.

**Systems Network Architecture (SNA).** The IBM architecture that defines the logical structure, formats, protocols, and operational sequences for transmitting information units through, and controlling the configuration and operation of, networks. The layered structure of SNA allows the ultimate origins and destinations of information, that is, the users, to be independent of and unaffected by the specific SNA network services and facilities used for information exchange.

# T

**tape volume set.** The collection of tape volumes on which a multivolume data set resides. A volume set is represented in one RACF profile.

**tape volume table of contents (TVTOC).** Information about a tape data set that RACF stores in the tape volume profile for the volume on which the data set resides. The TVTOC includes the data set name, data set sequence number, creation date, and an indicator as to whether a discrete tape data set profile exists.

**target node.** An RRSF node that a given RRSF node is logically connected to, as a result of a TARGET command. See *local node* and *remote node*.

**target user ID.** The target half of a source user ID and target user ID pair that has an established user ID association between them. For command direction, the target user ID is the user ID specified on the AT or ONLYAT keyword, and is the user ID under whose authority the command is run on the specified node. For password synchronization, the target user ID is the user ID whose password RACF automatically updates when the password for the source user ID is changed. Contrast with *source user ID*.

**task.** A basic unit of work to be performed or a process and the procedures that run the process.

**template.** Contains mappings of the profiles on the RACF database.

**TOKENBLD request.** The issuing of the RACROUTE macro with REQUEST=TOKENBLD specified. A TOKENBLD request builds a UTOKEN.

**TOKENMAP request.** The issuing of the RACROUTE macro with REQUEST=TOKENMAP specified. A TOKENMAP request maps a token in either internal or external format, allowing a caller to access individual fields within the UTOKEN.

**TOKENXTR request.** The issuing of the RACROUTE macro with REQUEST=TOKENXTR specified. A TOKENXTR request extracts a UTOKEN from the current address space, task or a caller-specified ACEE.

**TP.** See *transaction program*.

**tranquility.** Keeping the security classification of a resource constant while it is in use; keeping the security classification of a user constant while active.

**transaction program (TP).** A program that processes transactions in an SNA network.

**TVTOC.** See *tape volume table of contents*.

# U

**UACC.** See *universal access authority*.

**UADS.** See *user attribute data set*.

**UID.** See *OS/390 UNIX user identifier (UID)*.

**universal access authority (UACC).** The default access authority that applies to a resource if the user or group is not specifically permitted access to the resource, unless the user is restricted. The universal access authority can be any of the access authorities.

**user.** A person who requires the services of a computing system.

**user attribute.** The extraordinary privileges, restrictions, and processing environments assigned to a user. The user attributes are SPECIAL, AUDITOR, CLAUTH, OPERATIONS, GRPACC, ADSP, and REVOKE.

**user attribute data set (UADS).** In TSO, a partitioned data set with a member for each authorized user. Each member contains the appropriate passwords, user identifications, account numbers, LOGON procedure names, and user characteristics that define the user.

**user certificate.** A type of certificate managed by RACF. See *digital certificate*.

**user data set.** A data set defined to RACF in which either the high-level qualifier of the data set name or the qualifier supplied by an installation exit routine is a RACF user ID.

**user ID.** A RACF user ID. A string of 1-8 alphanumeric characters that uniquely identifies a RACF user, procedure, or batch job to the system. For TSO users, the user ID cannot exceed 7 characters and must begin with an alphabetic, #, $, or @ character. The user ID is defined by a user profile in the RACF database and is used as the name of the profile.

**user ID association.** A relationship between two user IDs, established through the RACLINK command, which is required for command direction and password synchronization between the user IDs. See *peer user ID association* and *managed user ID association*.

**user identification.** See *user ID*.

**user identification and verification.** The acts of identifying and verifying a RACF-defined user to the system during logon or batch job processing. RACF identifies the user by the user ID and verifies the user by the password, PassTicket, verified digital certificate, DCE credentials, or operator identification card supplied during logon processing or the password supplied on a batch JOB statement.

**user name.** In RACF, 1–20 alphanumeric characters that represent a RACF-defined user. Contrast with *user ID*.

**user profile.** A description of a RACF-defined user that includes the user ID, user name, default group name, password, profile owner, user attributes, and other information. A user profile can include information for subsystems such as TSO and DFP.

**UTOKEN.** The RACF user security token. A UTOKEN is an encapsulation or representation of the security characteristics of a user. RACF assigns a UTOKEN to each user in the system. See *STOKEN* and *RTOKEN*.

# V

**verification.** See *user identification and verification*.

**VERIFY request.** The issuing of the RACROUTE macro with REQUEST=VERIFY specified. A VERIFY request is used to verify the authority of a user to enter work into the system. The VERIFY request replaces the RACINIT function.

**VERIFYX request.** The issuing of the RACROUTE macro with REQUEST=VERIFYX specified. A VERIFYX request verifies a user and builds a UTOKEN, and handles the propagation of submitter ID.

**Virtual Machine (VM).** (1) An operating system that appears to be at the exclusive disposal of the particular user, but whose functions are accomplished by sharing the resources of a real data processing system. (2) In VM/ESA, the operating system that represents the virtual processors, virtual storage, virtual devices, and virtual channel subsystem allocated to a single user. A virtual machine also includes any expanded storage dedicated to it.

**VM.** See *Virtual Machine*.

# W

**workspace data sets.** VSAM data sets used by RACF for queuing requests sent to and received from target nodes in an RRSF environment.

# Index

## Special Characters



## A



## B



## C



## D



## E



## F



## G



## I

## M



## N



## O



## P



## Q



## R



## S



## T



## U



## W

# Readers' Comments — We'd Like to Hear from You

**OS/390**
**SecureWay Security Server RACF**
**Callable Services**

**Publication No. GC28-1921-08**

**Overall, how satisfied are you with the information in this book?**

| | Very Satisfied | Satisfied | Neutral | Dissatisfied | Very Dissatisfied |
|---|---|---|---|---|---|
| Overall satisfaction | ☐ | ☐ | ☐ | ☐ | ☐ |

**How satisfied are you that the information in this book is:**

| | Very Satisfied | Satisfied | Neutral | Dissatisfied | Very Dissatisfied |
|---|---|---|---|---|---|
| Accurate | ☐ | ☐ | ☐ | ☐ | ☐ |
| Complete | ☐ | ☐ | ☐ | ☐ | ☐ |
| Easy to find | ☐ | ☐ | ☐ | ☐ | ☐ |
| Easy to understand | ☐ | ☐ | ☐ | ☐ | ☐ |
| Well organized | ☐ | ☐ | ☐ | ☐ | ☐ |
| Applicable to your tasks | ☐ | ☐ | ☐ | ☐ | ☐ |

**Please tell us how we can improve this book:**

Thank you for your responses. May we contact you? ☐ Yes ☐ No

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

Name _____ Address _____

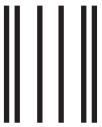
Company or Organization _____

Phone No. _____

**Readers' Comments — We'd Like to Hear from You**
GC28-1921-08

Cut or Fold
Along Line

Fold and Tape | **Please do not staple** | Fold and Tape



NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

# BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

IBM Corporation
Department 55JA, Mail Station P384
2455 South Road
Poughkeepsie, NY
12601-5400

Fold and Tape | **Please do not staple** | Fold and Tape



Cut or Fold
Along Line

Program Number: 5647-A01

Printed in the United States of America
on recycled paper containing 10%
recovered post-consumer fiber.

GC28-1921-08