z/VM
7.4

RACF Security Server Macros and Interfaces

SC24-6309-74

Note:

Before you use this information and the product it supports, read the information in "Notices" on page 443.

This edition applies to version 7, release 4 of IBM® z/VM® (product number 5741-A09) and to all subsequent releases and modifications until otherwise indicated in new editions.

Last updated: 2024-09-18

Contents

Figures

Tables

About This Document

This document describes how to customize and use macros provided with the IBM RACF® Security Server for z/VM.

Though this document is specific to z/VM, there are references to z/OS®. These references are applicable only when sharing a RACF database with a z/OS system, which is supported only on z/VM 7.2 and earlier versions.

This document contains a description (including syntax and related information) of macros provided with RACF. In addition, this document provides information on coding the interfaces used to invoke RACF from the RACF ISPF panels. It does not document the RACROUTE macro or the independent RACF system macros (such as RACHECK, RACDEF, and RACINIT) that are documented in *z/VM: Security Server RACROUTE Macro Reference*, GC28-1366-08.

Intended Audience

This document is intended for use by system programmers or installation personnel for:

- Installing RACF
- Maintaining RACF databases
- Writing, testing, and installing RACF exits
- Modifying the RACF program product to satisfy the installation's particular needs

Readers of this document should be familiar with the information in *z/VM: RACF Security Server General User's Guide*, *z/VM: RACF Security Server Security Administrator's Guide*, *z/VM: Security Server RACROUTE Macro Reference* and *z/VM: RACF Security Server System Programmer's Guide*.

z/VM: RACF Security Server Auditor's Guide, which describes using the RACF SMF data unload utility (RACFADU) and the RACF report writer, might also be useful.

Where to Find More Information

For information about related publications, refer to the "Bibliography" on page 447.

Links to Other Documents and Websites

The PDF version of this document contains links to other documents and websites. A link from this document to another document works only when both documents are in the same directory or database, and a link to a website works only if you have access to the Internet. A document link is to a specific edition. If a new edition of a linked document has been published since the publication of this document, the linked document might not be the latest edition.

How to provide feedback to IBM

We welcome any feedback that you have, including comments on the clarity, accuracy, or completeness of the information. See How to send feedback to IBM for additional information.

Summary of Changes for z/VM: RACF Security Server Macros and Interfaces

This information includes terminology, maintenance, and editorial changes. Technical changes or additions to the text and illustrations for the current edition are indicated by a vertical line (**|**) to the left of the change.

While IBM values the use of inclusive language, terms that are outside of IBM's direct influence, for the sake of maintaining user understanding, are sometimes required. As other industry leaders join IBM in embracing the use of inclusive language, IBM will continue to update the documentation.

SC24-6309-74, z/VM 7.4 (September 2024)

This edition supports the general availability of z/VM 7.4. Note that the publication number suffix (-74) indicates the z/VM release to which this edition applies.

SC24-6309-73, z/VM 7.3 (December 2023)

This edition includes terminology, maintenance, and editorial changes.

SC24-6309-73, z/VM 7.3 (September 2022)

This edition supports the general availability of z/VM 7.3. Note that the publication number suffix (-73) indicates the z/VM release to which this edition applies.

SC24-6309-02, z/VM Version 7.2 (July 2021)

This edition includes terminology, maintenance, and editorial changes.

SC24-6309-02, z/VM Version 7.2 (September 2020)

This edition supports the general availability of z/VM 7.2.

Chapter 1. RACF Customization Macros

This chapter contains Diagnosis, Modification or Tuning information intended to help the customer who uses RACF* product macros to customize a RACF installation.

Attention: Do not use this Diagnosis, Modification or Tuning Information as a programming interface.

This chapter describes the following macros that are available for use by your installation.

- **"ICHERCDE Macro" on page 1**—used to generate entries for the resource class descriptor table.
- **Appendix C, "ICHNCONV Macro," on page 341**—used to create the installation's naming convention table.
- **"ICHRFRTB Macro" on page 6**—used to generate entries in the RACF router table.
- **"GLBLDSK Macro" on page 10**—used on z/VM to create a list of minidisks in a global minidisk table.
- **"ICHNGMAX Macro" on page 13**—used on z/VM to specify the value for the POSIX constant NGROUPS_MAX.
- **"RACSERV Macro" on page 14**—used on z/VM to define the name of a RACF service machine.
- **"SYSSEC Macro" on page 14**—used to establish the relationship between RACF's response to access requests and the final disposition of the requests by z/VM.

For the descriptions and functions of the ICHEINTY, ICHETEST, and ICHEACTN product macros that are used to locate, update, test, and retrieve various profiles in the RACF database see Appendix B, "ICHEINTY, ICHETEST, and ICHEACTN Macros," on page 311. Because of the complexity of these macros and the cautions required in their use, IBM recommends the use of the RACROUTE REQUEST=EXTRACT system macro instead. See *z/VM: Security Server RACROUTE Macro Reference*.

ICHERCDE Macro

The ICHERCDE macro generates entries for the resource class descriptor table. The class descriptor table contains information that directs the processing of general resources. The table consists of an entry for each class except USER, GROUP, and DATASET. To generate the table, you must invoke the macro once for each class. To identify the end of the class descriptor table, you invoke the macro without specifying any operands.

The class descriptor table has two parts. The installation optionally supplies ICHRRCDE. ICHRRCDX and ICHRRCDE reside in RACFLINK LOADLIB. Refer to *z/VM: RACF Security Server System Programmer's Guide* for instructions on how to create ICHRRCDE.

Note:

1. Any installation planning to use RACROUTE to process classes which the installation has added, must either
   - code an ICHRFRTB macro instruction for each entry added to the class descriptor table to be accessed by the RACROUTE macro instruction, or
   - specify DECOUPL=YES on the RACROUTE macro instruction itself

   in order for RACROUTE to process the added classes. See "ICHRFRTB Macro" on page 6.
2. When you add a user-defined entry to the class descriptor table, you may need to include corresponding entries in the RACF router table as follows:
   a. If the class has RACLIST=ALLOWED, use ICHRFRTB to create a router table entry. Code the ICHRFRTB macro with ACTION=RACF and specify blanks for both the REQSTOR= and the SUBSYS= parameters so that RACF can process the class if you issue a SETROPTS RACLIST command.

b. If your application uses RACROUTE, and requires REQSTOR= and SUBSYS=, you will need to code an additional ICHRFRTB macro specifying ACTION=RACF and the appropriate requester and subsystem values.

3. A maximum of 1024 classes can be defined in the class descriptor table; 256 of these are reserved for use by IBM, leaving 768 for customer use. There are 1024 POSIT values, of which numbers 19–56 and 128–527 are available for installation use. Numbers 0–18, 57–127, and 528–1023 are reserved for IBM's use.

The ICHERCDE macro produces a CSECT for each invocation. If the CLASS operand is present, the CSECT name is the name of the class being defined; otherwise, the CSECT name is ICHRRCDE.

The ICHERCDE macro definition is as follows:

```
[label] ICHERCDE [CLASS=class-name]
                 [,DFTRETC=0|4|8]
                 [,DFTUACC=ALTER|CONTROL|UPDATE|READ|NONE]
                 [,FIRST=ALPHA|NUMERIC|ALPHANUM|ANY|
                    NONATABC|NONATNUM]
DISALLOWED]
                 [,GENLIST=ALLOWED|DISALLOWED]
                 [,GROUP=group-class|MEMBER=member-class]
                 [,ID=number]
                 [,KEYQUAL=0|nnn]
                 [,MAXLNTH=8|number]
                 [,OPER=YES|NO]
                 [,OTHER=ALPHA|NUMERIC|ALPHANUM|ANY|
                    NONATABC|NONATNUM]
                 [,POSIT=number]
                 [,PROFDEF=YES|NO]
                 [,RACLIST=ALLOWED|DISALLOWED]
                 [,RACLREQ=YES|NO]
                 [,RVRSMAC=YES|NO]
                 [,SLBLREQ=YES|NO]
```

CLASS=class-name

Specifies the name of the resource class. The name must be 4 to 8 characters long and must consist of the following: A through Z, 0 through 9, or # (X'7B'), @ (X'7C'), $ (X'5B'). The first character must be A through Z, # (X'7B'), @ (X'7C'), or $ (X'5B'). Installations must include a # (X'7B'), @ (X'7C'), $ (X'5B') or numeric character in the name of any class they define in order to guarantee that installation-defined classes do not conflict with IBM-defined classes. In this way, IBM-defined classes should always have unique class names. If this rule is not followed, the assembler issues a severity 4 MNOTE warning.

If you specify any options on the ICHERCDE macro, you must specify the CLASS operand.

DFTRETC=0|4|8

Specifies that the installation can decide, on a class by class basis, what the return code will be from a RACROUTE REQUEST=AUTH when RACF and the class are active and the class is RACLISTed (if RACLISTing is required by the CDT), but a profile does not exist for the resource that is being accessed.

- 0—the access request was accepted
- 4—no profile exists
- 8—the access request was denied

If you do not specify the parameter, it defaults to 4.

DFTUACC= ALTER|CONTROL|UPDATE|READ|NONE

Specifies the allowed access when you do not specify any access when defining resources within the class. If you omit DFTUACC, RACF uses the default universal access in the user's ACEE.

FIRST=

Specifies a character type restriction for the first character of the profile name.

ALPHA

Specifies an alphabetic or # (X'7B') @ (X'7C') $ (X'5B'). ALPHA is the default value for both the FIRST and OTHER operand.

NUMERIC
Specifies a digit (0–9).

ALPHANUM
Specifies an alphabetic, a numeric, or # (X'7B') @ (X'7C') $ (X'5B').

ANY
Specifies any character other than a blank, a comma, or a semicolon.

NONATABC
Specifies an alphabetic character. Characters such as # (X'7B') @ (X'7C') $ (X'5B') and numerics are excluded.

NONATNUM
Specifies an alphabetic or numeric character. # (X'7B') @ (X'7C') $ (X'5B') characters are excluded.

GENLIST=ALLOWED|DISALLOWED
Specifies whether SETROPTS GENLIST is to be allowed for the class. If you GENLIST the class on the SETROPTS command, then if a user requests access to a resource protected by a generic profile, a copy of that profile will be brought into the common storage area, rather than into the user's address space. RACF uses those generic profiles in common storage to check the authorization of any users who want to access the resource. The profiles remain in common storage until a REFRESH occurs.

GROUP=group-class
Specifies the name of the class that groups the resources within the class specified by the CLASS operand. If you omit this operand, RACF does not allow resource grouping for the resource specified by the CLASS operand. If group is specified, the group entry must be in the same class descriptor table (CDT) as the member entry (IBM or installation).

ID=number
Specifies a number from 1 to 255 that is associated with the class name. RACF stores this number in the general profile. Numbers 1 through 127 are reserved for use by IBM; numbers 128 through 255 are reserved for use by the installation.

The ID keyword need not be unique for each class; in fact, if more than 128 class descriptor table entries are defined by the installation, ID numbers will have to be reused. An installation can use ID numbers to identify related classes; however, RACF does not use the ID number. Do not confuse the ID number with the POSIT number described below.

If you specify any options on the ICHERCDE macro, you must specify the ID operand.

Note: If you change the value of ID for an existing class, you may get misleading messages from IRRUT200, since the change is made for the ID value in the class descriptor table, but not within the existing profiles. This could cause the wrong profiles to be associated with a specified class.

KEYQUAL=nnn
Specifies the number of matching qualifiers RACF uses when loading generic profile names to satisfy an authorization request if a discrete profile does not exist for the resource. Thus, if you specify two for the class, all generic profile names whose two highest level qualifiers match the two highest qualifiers of the entity name, are loaded into the RACF service machine's virtual storage, when the user requests access to a resource. If you do not specify KEYQUAL, the default is 0, and profile names for the entire class are loaded and searched. The maximum value you can specify for KEYQUAL is 123, which is the maximum number of qualifiers in a name 246 characters long.

MAXLNTH=8|number
Specifies the maximum length of names of resources defined by the CLASS operand. For installation-defined classes, you can specify a number from 1 to 246; the default is 8.

Note: You cannot use the MAXLNTH parameter to change the maximum size allowed for a resource name by the resource manager. For example, CICS® allows a maximum of 13 characters in a transaction name. Thus, if you define additional CICS transaction classes, you must also specify MAXLNTH=13.

MEMBER=member-class
Specifies the name of the class grouped by the resources within the class specified by the CLASS operand. The class name must be from 1 to 8 alphanumeric characters. When this operand is specified, the class being defined is a resource group. If a member is specified, the member entry must be in the same CDT (IBM or installation) as the group entry.

OPER=YES|NO
Specifies whether RACF is to take the OPERATIONS attribute into account when it performs authorization checking. If YES is specified, RACF considers the OPERATIONS attribute; if NO is specified, RACF ignores the OPERATIONS attribute. YES is the default.

OTHER=
Specifies a character type restriction for the characters of the profile name other than the first character.

ALPHA
Specifies an alphabetic or # (X'7B') @ (X'7C') $ (X'5B'). ALPHA is the default value for both the FIRST and OTHER operand.

NUMERIC
Specifies a digit (0-9).

ALPHANUM
Specifies an alphabetic, numeric, or # (X'7B') @ (X'7C') $ (X'5B').

ANY
Specifies any character other than a blank, comma, or semicolon.

NONATABC
Specifies an alphabetic character. Characters such as # (X'7B') @ (X'7C') $ (X'5B') and numerics are excluded.

NONATNUM
Specifies an alphabetic or numeric character. # (X'7B') @ (X'7C') $ (X'5B') characters are excluded.

POSIT=number
Each class in the class descriptor table has a POSIT number specified on the ICHERCDE macro. The POSIT number identifies a set of option flags that controls the following RACF processing options:

- Status of authorization checking for the class (Active/Inactive)
- Whether auditing should take place for resources within the class
- Whether statistics should be kept for resources within the class
- Whether generic profile access checking is active for the class
- Whether generic command processing is active for the class
- Whether global access checking is active for the class
- Whether user has CLAUTH to a resource class
- Whether special resource access auditing applies to the class (SETROPTS LOGOPTIONS)
- Whether SETROPTS RACLIST will occur for this class (when the parameter RACLIST=ALLOWED is also coded).

Before you assemble the CDT, you must decide whether to use a unique set of option flags for each RACF class or whether to have two or more RACF classes share the same set of option flags. It is not recommended that a RACF class that has a default return code of 8 share a POSIT value with a RACF class having a default return code not equal to 8. If a class with a default return code of 8 is activated but no profiles are defined, user activity that requires access in that class will be prevented.

If you choose to use a unique set of option flags for a class, assign the class a unique POSIT number. If you choose to share the same set of option flags among several classes, assign those classes the same POSIT number. After creating your class descriptor table, you can activate the classes that comprise it and their respective set of option flags via the appropriate keywords on the SETROPTS command.

There are 1024 POSIT numbers that can identify 1024 sets of option flags. Installations can specify POSIT numbers 1–56 and 128–527. Numbers 0–18, 57–127, and 528–1023 are reserved for IBM's use.

The following is an example of the use of POSIT numbers:

You decide to define the class of $PONIES. When you code the ICHERCDE macro, you can specify the class as $PONIES and specify the POSIT number as 21. (See POSIT Numbers diagram.) In this case, you decided that 21 will represent "Auditing for resources within the class" and "Statistics for resources within the class."

Later, you decide to define the class of $HORSES, a class ‘related’ to $PONIES, and logically requiring the same RACF processing options. Therefore, when you code the ICHERCDE macro to include the $HORSES class in the class descriptor table, specify the POSIT number as 21, the same as for $PONIES.

You can then assemble and link-edit the macros to create the installation portion of the CDT. After the CDT is established, you must activate each class and the options you have chosen for that class (in this case, auditing for resources within the class and statistics for resources within the class) by issuing the SETROPTS command:

```
SETROPTS  CLASSACT($PONIES) STATISTICS($PONIES) AUDIT($PONIES)
```

Because you have specified the same posit number for both $PONIES and $HORSES (the classes share the same option flag), you do not need to reissue the command to activate the same set of options for $HORSES. RACF does it automatically because a relationship has been established between the POSIT number (on the ICHERCDE macro) and the set of options it represents (activated on the SETROPTS command.)

Be aware that if two or more classes share the same POSIT number, and you make a change to the option flag set of one of the classes via the SETROPTS command, the change will also be in effect for all the classes that share that POSIT number. Thus, if you turn off statistics options for the class of $PONIES, that action turns off statistics for the class of $HORSES because both classes share the same POSIT number. You must code a unique POSIT number for each class if you want RACF to independently control processing options.

If you change the POSIT value, follow up with the SETROPTS LIST command, since changing the POSIT value could cause unpredictable results. For example, you could deactivate a class if you change it to use a POSIT value associated with a class that is not active.

Note: Your installation must include a # (X'7B') @ (X'7C') or $ (X'5B') or numeric character in the name of any class your installation chooses to define.

PROFDEF=YES|NO
Specifies whether you want RACF to allow profiles to be defined for this RACF resource class. If you specify PROFDEF=NO, RACF will not allow profiles to be defined to this RACF resource class; if a user attempts to define a profile to that class, the RDEFINE command responds with an appropriate message.

RACLIST=ALLOWED|DISALLOWED
Specifies whether SETROPTS RACLIST is to be allowed for the class. If you RACLIST the class on the SETROPTS command, RACF brings copies of all discrete and generic profiles within that class into common storage. RACF uses those profiles in common storage to check the authorization of any users who want to access the resources. The profiles remain in common storage until a REFRESH occurs.

RACLREQ=YES|NO
Specifies whether you must have RACLISTed the class, either via the RACLIST macro or via SETROPTS RACLIST, in order to have a RACHECK performed. The purpose of this keyword is to allow routines that cannot tolerate I/O, to invoke RACF. If you specify YES, and the class is not RACLISTed and a RACROUTE REQUEST=AUTH is attempted, the return code is 4. If you do not specify the parameter, it defaults to NO.

RVRSMAC=YES|NO
Specifies whether reverse mandatory access checking is required.

If RVRSMAC=YES is specified, RACF performs a reverse mandatory access check (MAC) when and if a mandatory access check is required. In a reverse mandatory access check, the SECLABEL of the resource must dominate that of the user.

Note that if this parameter is omitted, it is assigned the default value of RVRSMAC=NO, which means that when and if a mandatory access check is required, the user's SECLABEL must dominate that of the resource.

SLBLREQ=YES|NO
Specifies whether SECLABEL is required for the profiles of this class.

When MLACTIVE is on, each profile in the class must have a SECLABEL. The default, SLBLREQ=NO, means that RACF will not require a SECLABEL for profiles in this class; however, if a SECLABEL exists for this profile, and the SECLABEL class is active, RACF will use it during authorization checking.

SLBLREQ=NO applies to general resource classes that have no profiles, such as DIRAUTH, or for classes that contain no data, such as OPERCMDS and SECLABEL).

ICHRFRTB Macro

The ICHRFRTB macro generates entries in the RACF router table. This table controls the action taken by the RACF router ICHRFR00 when invoked by the RACROUTE macro. The router table has two parts. IBM supplies module ICHRFR0X, which must not be modified. The installation optionally supplies module ICHRFR01 so you can add entries for locally defined resource classes or requestor/subsystem combinations.

The IBM-supplied ICHRFR0X module contains one entry for each entry in the class descriptor table, plus one entry each for DATASET, USER, GROUP, and CONNECT. For all entries, the operands REQSTOR and SUBSYS have the default value (all blanks), and the ACTION operand is set to RACF. In addition, several special entries related to program control and tape data set support are present. These entries also have non-blank REQSTOR and SUBSYS values.

When you add a user-defined entry to the class descriptor table, you may need to include corresponding entries in the RACF router table as follows:

1. If the class has RACLIST=ALLOWED, use ICHRFRTB to create a router table entry. Code the ICHRFRTB macro with ACTION=RACF and specify blanks for both the REQSTOR= and the SUBSYS= parameters so that RACF can process the class if you issue a SETROPTS RACLIST command.
2. If your application uses RACROUTE, and requires REQSTOR= and SUBSYS=, you will need to code an additional ICHRFRTB macro specifying ACTION=RACF and the appropriate requester and subsystem values.

ICHRFRTB concatenates the values specified for the REQSTOR, SUBSYS, and CLASS operands to form a 24-character string defining the entry. The macro matches these values against the string formed by the values specified on the RACROUTE macro instruction.

The ICHRFRTB macro definition is as follows:

```
[label] ICHRFRTB [ACTION=NONE|RACF]
                 [,CLASS=class-name]
                 [,REQSTOR=requestor-name]
                 [,SUBSYS=subsystem-name]
                 [TYPE=END]
```

ACTION=
Specifies the action to be taken for this entry. This operand is required unless TYPE=END is specified.

NONE — specifies that no action is to be taken for this entry.

RACF — specifies that RACF is to be called for this entry.

CLASS=class-name
specifies the name of the resource class. You must use the same name that is specified in the corresponding class descriptor table entry. This operand is required unless TYPE=END is specified.

REQSTOR=requestor-name
Specifies the 8-character name to be used, along with CLASS and SUBSYS, to form the 24-character string defining the entry. Installations should begin requestor names with a # (X'7B'), @ (X'7C') or $ (X'5B'), because IBM-defined requestor names do not begin with those characters. If you do not specify a requestor name, the default is a string of 8 blanks. If you code REQSTOR, you should also code the CLASS operand.

SUBSYS=subsystem-name
Specifies the 8-character name to be used, along with CLASS and REQSTOR, to form the 24-character string defining the entry. Installations should begin subsystem names with a # (X'7B'), @ (X'7C') or $ (X'5B'), because IBM-defined subsystem names will not begin with such characters. If no subsystem name is specified, it defaults to a string of 8 blanks. This operand should not be coded unless CLASS is also specified.

TYPE=END
Indicates the end of the ICHRFR01 table. You must code TYPE=END on the last ICHRFRTB macro instruction. If TYPE=END is specified, no other operands can be coded.

The RACF Router Table Supplied by IBM (ICHRFR0X)

```
ICHRFRTB CLASS=DATASET,ACTION=RACF
ICHRFRTB CLASS=USER,ACTION=RACF
ICHRFRTB CLASS=GROUP,ACTION=RACF
ICHRFRTB CLASS=CONNECT,ACTION=RACF
ICHRFRTB CLASS=DASDVOL,ACTION=RACF
ICHRFRTB CLASS=GDASDVOL,ACTION=RACF
ICHRFRTB CLASS=TAPEVOL,ACTION=RACF
ICHRFRTB CLASS=TERMINAL,ACTION=RACF
ICHRFRTB CLASS=GTERMINL,ACTION=RACF
ICHRFRTB CLASS=APPL,ACTION=RACF
ICHRFRTB CLASS=TIMS,ACTION=RACF
ICHRFRTB CLASS=GIMS,ACTION=RACF
ICHRFRTB CLASS=AIMS,ACTION=RACF
ICHRFRTB CLASS=TCICSTRN,ACTION=RACF
ICHRFRTB CLASS=GCICSTRN,ACTION=RACF
ICHRFRTB CLASS=PCICSPSB,ACTION=RACF
ICHRFRTB CLASS=QCICSPSB,ACTION=RACF
ICHRFRTB CLASS=GMBR,ACTION=RACF
ICHRFRTB CLASS=GLOBAL,ACTION=RACF
ICHRFRTB CLASS=DSNR,ACTION=RACF
ICHRFRTB CLASS=FACILITY,ACTION=RACF
ICHRFRTB CLASS=SCDMBR,ACTION=RACF
ICHRFRTB CLASS=SECDATA,ACTION=RACF
ICHRFRTB CLASS=FCICSFCT,ACTION=RACF
ICHRFRTB CLASS=HCICSFCT,ACTION=RACF
ICHRFRTB CLASS=JCICSJCT,ACTION=RACF
ICHRFRTB CLASS=KCICSJCT,ACTION=RACF
ICHRFRTB CLASS=DCICSDCT,ACTION=RACF
```

```
ICHRFRTB CLASS=ECICSDCT,ACTION=RACF
ICHRFRTB CLASS=SCICSTST,ACTION=RACF
ICHRFRTB CLASS=UCICSTST,ACTION=RACF
ICHRFRTB CLASS=MCICSPPT,ACTION=RACF
ICHRFRTB CLASS=NCICSPPT,ACTION=RACF
ICHRFRTB CLASS=ACICSPCT,ACTION=RACF
ICHRFRTB CLASS=BCICSPCT,ACTION=RACF
ICHRFRTB CLASS=PMBR,ACTION=RACF
ICHRFRTB CLASS=PROGRAM,ACTION=RACF
ICHRFRTB CLASS=DATASET,REQSTOR=CLOSE,SUBSYS=OCEOV,ACTION=RACF
ICHRFRTB CLASS=DATASET,REQSTOR=TAPEOPEN,SUBSYS=OCEOV,ACTION=RACF
ICHRFRTB CLASS=DATASET,REQSTOR=TAPERST,SUBSYS=RESTART,ACTION=RACF
ICHRFRTB CLASS=TAPEVOL,REQSTOR=TAPEOPEN,SUBSYS=OCEOV,ACTION=RACF
ICHRFRTB CLASS=TSOPROC,ACTION=RACF
ICHRFRTB CLASS=ACCTNUM,ACTION=RACF
ICHRFRTB CLASS=PERFGRP,ACTION=RACF
ICHRFRTB CLASS=TSOAUTH,ACTION=RACF
ICHRFRTB CLASS=DATASET,REQSTOR=TAPEEOV,SUBSYS=OCEOV,ACTION=RACF
ICHRFRTB CLASS=TAPEVOL,REQSTOR=TAPEEOV,SUBSYS=OCEOV,ACTION=RACF
ICHRFRTB CLASS=VMCMD,ACTION=RACF
ICHRFRTB CLASS=VMNODE,ACTION=RACF
ICHRFRTB CLASS=VMBATCH,ACTION=RACF
ICHRFRTB CLASS=FIELD,ACTION=RACF
ICHRFRTB CLASS=PROPCNTL,ACTION=RACF
ICHRFRTB CLASS=PROPCNTL,REQSTOR=PROPCHK,SUBSYS=IEFCMAUT,ACTION=RACF
ICHRFRTB CLASS=MGMTCLAS,ACTION=RACF
ICHRFRTB CLASS=STORCLAS,ACTION=RACF
ICHRFRTB CLASS=FACILITY,REQSTOR=ABDUMP,SUBSYS=ABDUMP,ACTION=RACF
ICHRFRTB CLASS=VMBR,ACTION=RACF
ICHRFRTB CLASS=VMEVENT,ACTION=RACF
ICHRFRTB CLASS=VXMBR,ACTION=RACF
ICHRFRTB CLASS=VMXEVENT,ACTION=RACF
ICHRFRTB CLASS=VMMDISK,ACTION=RACF
ICHRFRTB CLASS=VMRDR,ACTION=RACF
ICHRFRTB CLASS=APPCLU,ACTION=RACF
ICHRFRTB CLASS=SECLABEL,ACTION=RACF
ICHRFRTB CLASS=SMESSAGE,ACTION=RACF
ICHRFRTB CLASS=DEVICES,ACTION=RACF
ICHRFRTB CLASS=VTAMAPPL,ACTION=RACF
ICHRFRTB CLASS=PSFMPL,ACTION=RACF
ICHRFRTB CLASS=OPERCMDS,ACTION=RACF
```

```
ICHRFRTB CLASS=WRITER,ACTION=RACF
ICHRFRTB CLASS=JESSPOOL,ACTION=RACF
ICHRFRTB CLASS=JESJOBS,ACTION=RACF
ICHRFRTB CLASS=JESINPUT,ACTION=RACF
ICHRFRTB CLASS=CONSOLE,ACTION=RACF
ICHRFRTB CLASS=TEMPDSN,ACTION=RACF
ICHRFRTB CLASS=DIRAUTH,ACTION=RACF
ICHRFRTB CLASS=SURROGAT,ACTION=RACF
ICHRFRTB CLASS=NODES,ACTION=RACF
ICHRFRTB CLASS=NODMBR,ACTION=RACF
ICHRFRTB CLASS=RVARSMBR,ACTION=RACF
ICHRFRTB CLASS=RACFVARS,ACTION=RACF
ICHRFRTB CLASS=CIMS,ACTION=RACF
ICHRFRTB CLASS=DIMS,ACTION=RACF
ICHRFRTB CLASS=DLFCLASS,ACTION=RACF
ICHRFRTB CLASS=CCICSCMD,ACTION=RACF
ICHRFRTB CLASS=VCICSCMD,ACTION=RACF
ICHRFRTB CLASS=PIMS,ACTION=RACF
ICHRFRTB CLASS=QIMS,ACTION=RACF
ICHRFRTB CLASS=SIMS,ACTION=RACF
ICHRFRTB CLASS=UIMS,ACTION=RACF
ICHRFRTB CLASS=FIMS,ACTION=RACF
ICHRFRTB CLASS=HIMS,ACTION=RACF
ICHRFRTB CLASS=OIMS,ACTION=RACF
ICHRFRTB CLASS=WIMS,ACTION=RACF
ICHRFRTB CLASS=VMMAC,ACTION=RACF
ICHRFRTB CLASS=VMSEGMT,ACTION=RACF
ICHRFRTB CLASS=SDSF,ACTION=RACF
ICHRFRTB CLASS=GSDSF,ACTION=RACF
ICHRFRTB CLASS=APPCTP,ACTION=RACF
ICHRFRTB CLASS=APPCSI,ACTION=RACF
ICHRFRTB CLASS=APPCPORT,ACTION=RACF
ICHRFRTB CLASS=CSFSERV,ACTION=RACF
ICHRFRTB CLASS=CSFKEYS,ACTION=RACF
ICHRFRTB CLASS=GCSFKEYS,ACTION=RACF
ICHRFRTB CLASS=NVASAPDT,ACTION=RACF
ICHRFRTB CLASS=PROGRAM,REQSTOR=PROGMCHK,SUBSYS=CONTENTS,ACTION=RACF
ICHRFRTB CLASS=USER,REQSTOR=FMH5-MGR,SUBSYS=APPC/z/OS,ACTION=RACF
ICHRFRTB CLASS=USER,REQSTOR=VTAMEXIT,SUBSYS=APPC/z/OS,ACTION=RACF
ICHRFRTB CLASS=USER,REQSTOR=SIGNONTP,SUBSYS=APPC/z/OS,ACTION=RACF
ICHRFRTB CLASS=USER,REQSTOR=APPCSCH,SUBSYS=APPC/z/OS,ACTION=RACF
```

```
ICHRFRTB CLASS=APPCTP,REQSTOR=APPCSDFM,SUBSYS=APPC/z/OS,ACTION=RACF
ICHRFRTB CLASS=APPCSI,REQSTOR=APPCSDFM,SUBSYS=APPC/z/OS,ACTION=RACF
ICHRFRTB CLASS=FACILITY,REQSTOR=APPCSDFM,SUBSYS=APPC/z/OS,ACTION=RACF
ICHRFRTB CLASS=RMTOPS,ACTION=RACF
ICHRFRTB CLASS=PROGRAM,REQSTOR=PADSCHK,SUBSYS=CONTENTS,ACTION=RACF
ICHRFRTB CLASS=INFOMAN,ACTION=RACF
ICHRFRTB CLASS=GINFOMAN,ACTION=RACF
ICHRFRTB CLASS=APPCSERV,ACTION=RACF
ICHRFRTB CLASS=APPCSERV,REQSTOR=APPCSDFM,SUBSYS=APPC/z/OS,ACTION=RACF
ICHRFRTB CLASS=PTKTDATA,ACTION=RACF
ICHRFRTB CLASS=LFSCLASS,ACTION=RACF
ICHRFRTB CLASS=RODMMGR,ACTION=RACF
ICHRFRTB CLASS=MQQUEUE,ACTION=RACF
ICHRFRTB CLASS=GMQQUEUE,ACTION=RACF
ICHRFRTB CLASS=MQPROC,ACTION=RACF
ICHRFRTB CLASS=GMQPROC,ACTION=RACF
ICHRFRTB CLASS=MQNLIST,ACTION=RACF
ICHRFRTB CLASS=GMQNLIST,ACTION=RACF
ICHRFRTB CLASS=MQADMIN,ACTION=RACF
ICHRFRTB CLASS=GMQADMIN,ACTION=RACF
ICHRFRTB CLASS=MQCMDS,ACTION=RACF
ICHRFRTB CLASS=MQCONN,ACTION=RACF
ICHRFRTB CLASS=DIRSRCH,ACTION=RACF
ICHRFRTB CLASS=DIRACC,ACTION=RACF
ICHRFRTB CLASS=FSOBJ,ACTION=RACF
ICHRFRTB CLASS=FSSEC,ACTION=RACF
ICHRFRTB CLASS=PROCESS,ACTION=RACF
ICHRFRTB CLASS=VMPOSIX,ACTION=RACF
ICHRFRTB CLASS=DIRECTRY,ACTION=RACF
ICHRFRTB CLASS=FILE,ACTION=RACF
ICHRFRTB CLASS=SFSCMD,ACTION=RACF
ICHRFRTB CLASS=VMLAN,ACTION=RACF
ICHRFRTB CLASS=CBIND,ACTION=RACF
ICHRFRTB CLASS=SERVAUTH,ACTION=RACF
ICHRFRTB CLASS=XFACILIT,ACTION=RACF
ICHRFRTB CLASS=GXFACILI,ACTION=RACF
ICHRFRTB CLASS=RACFEVNT,ACTION=RACF
```

GLBLDSK Macro

The GLBLDSK macro creates a list of minidisks in a global minidisk table. The table can grant READ access to these minidisks, referred to as *public minidisks*. Use the following to identify public minidisks:

- They have no installation-sensitive data on them
- They are linked in R or RR mode
- All users are authorized to read the data on them
- They require no auditing.

  Note: If ACIGROUPs are used on your system, the minidisk may be defined as public within the scope of an ACIGROUP.

When a user attempts to link to a minidisk in R or RR mode, RACF first searches the global minidisk table. If the READ link request is satisfied by the global minidisk table, the RACF service machine is not called for an access decision. Because no call is made to the RACF service machine, any profiles that might disallow access to a minidisk you are protecting are not considered.

When a link request is satisfied by the global minidisk table, no auditing of any type is done and no profile options (such as NOTIFY, SECLABEL, SECLEVEL, or CATEGORIES) are used.

If the request cannot be satisfied by the global minidisk table, the request is sent to the RACF service machine for normal RACF processing.

To avoid a security exposure of a sensitive minidisk, do not create an entry in the global minidisk table for a minidisk protected by a profile containing a security level, security category, or security label. (If the security label in the profile is SYSLOW, a global minidisk table entry can be created.)

The GLBLDSK macro must be coded in the RACF module HCPRWA. You add a GLBLDSK entry to the HCPRWA module for every minidisk you identify as public. Create an update file for HCPRWA to do this. Each time you add, change, or delete GLBLDSK macro entries, you must change the update file, reassemble HCPRWA, and regenerate the CP nucleus. Be sure that RACF MACLIB is included in your MACLIB concatenation when you assemble HCPRWA. For instructions on performing this local modification to HCPRWA (a CP source part), see *z/VM: Service Guide*.

You may have multiple GLBLDSK entries for one minidisk (see “Example 2” on page 12). The maximum number of GLBLDSK entries that can be specified in HCPRWA is 30.

The syntax of the GLBLDSK macro follows; the defaults are underlined.

```
[label] GLBLDSK USERID=userid
         ,VADDR=virtual_address
        [,ACIGRP=acigroup_name]
        [,SCOPE=LOCAL|GLOBAL]
        [,LAST=NO|YES]
```

USERID=*userid*
Must be the 1-to-8-character user ID of the minidisk owner.

VADDR=*virtual_address*
Must be the 1-to-4-character virtual address of the disk.

ACIGRP=*acigroup_name*
Allows members of an additional ACIGROUP access to the disk. It is not necessary for all users to belong to an ACIGROUP. The owner of the minidisk does not have to belong to an ACIGROUP. This keyword is optional.

SCOPE=LOCAL|GLOBAL
Determines whether the ACIGROUP is considered in authorization checking. This keyword is optional; not all users must have an ACIGROUP.

LOCAL
The ACIGROUP of the minidisk is used to determine if a user should be allowed to link in R or RR mode to the minidisk.

GLOBAL
The ACIGROUP of the disk is ignored when a user attempts to access the minidisk.

You cannot specify SCOPE=GLOBAL and ACIGRP keywords on the same GLBLDSK invocation.

LAST=NO|YES
LAST=NO is the default. LAST=YES must be specified on the last invocation of GLBLDSK in HCPRWA.

Example 1

The system programmer, working with the security administrator, identifies two minidisks to be made public:

- MAINT's 190 minidisk
- MAINT's 19E minidisk

There is no sensitive data on these two disks.

Assumptions

All CMS users require read access to these disks; ACIGROUPs are not used on the system.

Create an update file for HCPRWA with entries as follows:

```
GLBLDSK USERID=MAINT,VADDR=190
GLBLDSK USERID=MAINT,VADDR=19E,LAST=YES
```

Results

When a general user attempts to link in R or RR mode to the 190 or 19E minidisks owned by MAINT, the global minidisk table in HCPRWA is searched for a match. Since a match is found, access is granted.

Example 2

The system programmer, working with the security administrator, identifies two minidisks to be made public for two ACIGROUPs:

- MAINT's 19C minidisk
- MAINT's 19D minidisk

There is no sensitive data on these two disks. The ACIGROUPs on the system are SYSP1, PROGRM, and OTHER. The MAINT user ID *does not* belong to any ACIGROUP.

Assumptions

All CMS users in the SYSP1 and PROGM groups require read access to these disks.

Create an update file for HCPRWA with entries as follows:

```
GLBLDSK USERID=MAINT,VADDR=19C,ACIGRP=SYSP1
GLBLDSK USERID=MAINT,VADDR=19D,ACIGRP=SYSP1
GLBLDSK USERID=MAINT,VADDR=19C,ACIGRP=PROGM
GLBLDSK USERID=MAINT,VADDR=19D,ACIGRP=PROGM,LAST=YES
```

Results

USERA is a member of ACIGROUP SYSP1. When USERA attempts to link in R or RR mode to MAINT's 19C disk there is a match in the global minidisk table and access is granted.

USERB is a member of ACIGROUP PROGM. When USERB attempts to link in R or RR mode to MAINT's 19D disk there is a match in the global minidisk table and access is granted.

There is not an ACIGROUP entry in USERC's CP directory entry. USERC attempts to link in R or RR mode to MAINT's 19D disk, there is a match in the global minidisk table and access is granted. Remember that MAINT does not have an ACIGROUP entry. The user and MAINT therefore have the same ACIGROUP (none) so access is granted.

USERD is a member of ACIGROUP OTHER. When USERD attempts to link in R or RR mode to MAINT's 19D disk, no match is found and the access request is sent to a RACF service machine for processing.

To summarize: The 19C and 19D minidisks belonging to MAINT are public minidisks. Users that request read access to them must either:

- Belong to ACIGROUP PROGM, or
- Belong to ACIGROUP SYSP1, or
- Not have an ACIGROUP entry in their CP directory

Example 3

The system programmer, working with the security administrator, identifies one minidisk to be made public:

- MAINT's 190 minidisk

There is no sensitive data on this disk. ACIGROUPs are used on this system and MAINT belongs to the ACIGROUP named SUPRT.

Assumptions

All CMS users require read access to the disks.

Create an update file for HCPRWA with entries as follows:

```
GLBLDSK USERID=MAINT,VADDR=190,SCOPE=GLOBAL,LAST=YES
```

Results

The ACIGROUPs are not examined when using the global minidisk table to grant read access to any user.

ICHNGMAX Macro

Use the ICHNGMAX macro to specify the value for the POSIX constant NGROUPS_MAX. NGROUPS_MAX defines the maximum number of GIDs (the size of the supplementary group list) that are associated with a POSIX process for authorization purposes. As a POSIX standard, the NGROUPS_MAX value must remain constant throughout the life of any process. Therefore, you can only change this value when you IPL.

Also, a valid value on the ICHNGMAX macro triggers RACF to notify z/VM during RACF initialization that RACF supports OpenExtensions. z/VM uses the OpenExtensions information defined in the RACF database, rather than information in the CP directory. z/VM then calls RACF to authorize and audit security requests in an OpenExtensions environment.

By default, ICHNGMAX has a value of zero. With this default value, no error message is issued. If the zero value remains unchanged for ICHNGMAX, RACF does not inform z/VM that RACF supports OpenExtensions. In this case, OpenExtensions information contained in the RACF database is not used and z/VM does not call RACF to authorize and audit security requests in an OpenExtensions environment.

To specify a value for NGROUPS_MAX and enable RACF support for OpenExtensions on z/VM, change the zero value on the ICHNGMAX macro in the HCPRWA module.

If you code a value on the ICHNGMAX macro that is not valid, you receive an error message and assembly is ended. You must:

1. Code a valid value on ICHNGMAX
2. Reassemble the HCPRWA module
3. Regenerate the CP nucleus. Be sure that RACF MACLIB is included in your MACLIB concatenation when you assemble the HCPRWA module.

For instructions on performing this local modification to HCPRWA (a CP source part), see *z/VM: Service Guide*.

The syntax of the ICHNGMAX macro is as follows:

```
[label]  ICHNGMAX VALUE=value
```

VALUE=*value*
Must be an integer value between 32 and 125, inclusive.

Note:

1. Unless SETROPTS GRPLIST is in effect, a supplementary group list (SGID list) is not associated with OpenExtensions processes for any user. In this case, only the GID associated with the user's default RACF group is associated with processes created by the user.
2. When SETROPTS GRPLIST is in effect, RACF scans through the user's group list in alphabetical order searching for the first *value* number of unique GIDs associated with the groups in the user's group list. These GIDs will comprise the SGID list that is returned to z/VM for this user.
3. Database I/O is performed during LOGON/XAUTOLOG and also when SGID lists are requested by z/VM services. The number of I/Os equals the number of groups the user is connected to, up to the time that *value* unique GID values have been retrieved.

RACSERV Macro

Each RACSERV macro defines the name of a RACF service machine. You can define a maximum of 10 service machines.

To define a RACF service machine, create an update file for HCPRWA. The RACSERV macros must be coded immediately following the HCPRWATB entry definition label in HCPRWA and there must be one RACSERV macro coded for each RACF service machine being defined.

The

```
DC    X'FFFFFFFF'
```

instruction that appears in HCPRWA directly following the RACSERV macros is there to mark the end of the service machine list and must not be removed. If it is moved or deleted, unpredictable behavior will result when the service machine table is scanned during normal system operation. You must reassemble HCPRWA and regenerate the CP nucleus after adding the RACSERV macros. Be sure that RACF MACLIB is included in your MACLIB concatenation when you assemble HCPRWA. For instructions on performing this local modification to HCPRWA (a CP source part), see *z/VM: Service Guide*.

The syntax of the RACSERV macro is as follows:

```
[label]  RACSERV USERID=serverid[,CPUSE=YES|NO]
```

USERID=*serverid*
Where *serverid* is the user ID of the z/VM machine that is to act as a RACF service machine. The *serverid* must be alphanumeric and cannot contain more than 8 characters.

CPUSE=YES|NO
Indicates whether the service machine will process requests (for example, LINKs, LOGONs, and other CP commands) from the z/VM operating system that are reflected to the service machine. For example, specify CPUSE=NO if you want to dedicate a particular service machine to processing RACROUTE requests. CPUSE defaults to YES.

SYSSEC Macro

Access to system resources is managed by z/VM; however, when RACF is installed, the z/VM system resource manager calls RACF whenever a user requests access to a protected resource. Based on information contained in its various profiles and SYSSEC options, RACF responds to the resource manager, indicating whether the requested access is authorized or not.

The SYSSEC macro, part of the RACF module HCPRWA, establishes the relationship between RACF's response to an access request and the final disposition of that request by z/VM. RACF checks whether the access request is authorized. The SYSSEC macro options are then used to determine the final disposition of the request.

SYSSEC parameter options are supported for VMMDISK, VMRDR, VMNODE, VMCMD, and VMLAN classes. There is also a default category which covers resource checks made by CP using a generalized resource checking interface defined in ACIPARMS (function code ACIRSCHK). There are no SYSSEC options supported for VMMAC, VMSEGMT, or other general resource classes not covered by the categories listed above.

A summary of SYSSEC parameters and their default settings is shown in Figure 1 on page 15.

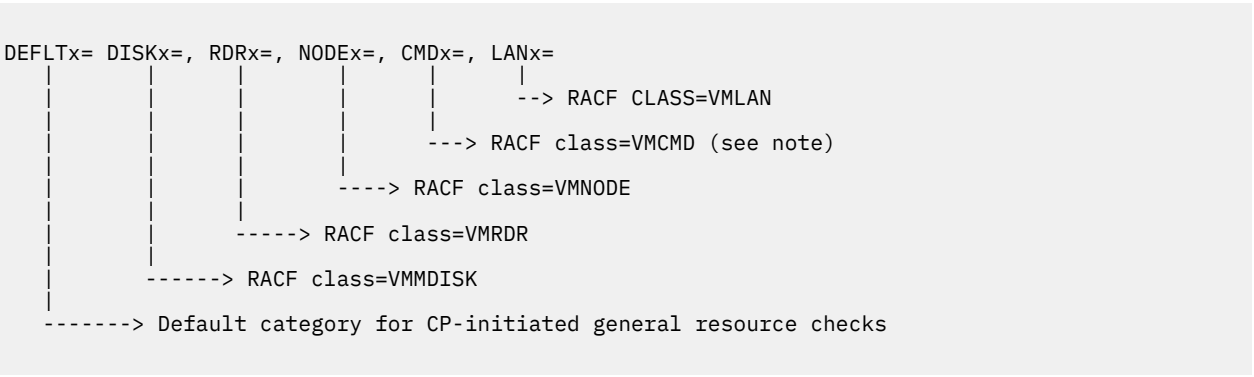


Figure 1. SYSSEC Options

Where *x* =	RACF Response to z/VM	SYSSEC Keyword Values
P	Authorized (PERMIT)	ALLOW | DEFER
F	Access Not Authorized (FAIL)	DEFER | FAIL
W	Not Authorized, WARNING	DEFER | FAIL
U	Resource Not Defined to RACF	ALLOW | DEFER | FAIL
M	Display RACF messages	ON | OFF

Note: SYSSEC parameter options are supported for the following commands and diagnose codes protected in the VMCMD class: STORE.C, TRSOURCE, DEFINE.MDISK, DIAG088, DIAG0E4, XAUTOLOG.G, DIAG0A0.VALIDATE, DIAG0A0.QUERYSEC, and DIAG0A0.HRTSTORE.

To change the defaults, create an update file for HCPRWA and modify the DISK*x*, RDR*x*, NODE*x*, CMD*x*, or LAN*x* parameters on the SYSSEC macro.

If you change the SYSSEC options, you must reassemble HCPRWA before generating the CP nucleus. Be sure that RACF MACLIB is included in your MACLIB concatenation when you assemble HCPRWA. For instructions on performing this local modification to HCPRWA (a CP source part), see *z/VM: Service Guide*.

The syntax of the SYSSEC macro follows; the defaults are underlined. Parameters shown within brackets [ ] are optional.

```
[label] SYSSEC [DEFLTP= ALLOW |DEFER]    X
               [,DISKP= ALLOW |DEFER]    X
               [,RDRP= ALLOW |DEFER]     X
               [,NODEP= ALLOW |DEFER]    X
               [,CMDP= ALLOW |DEFER]     X
               [,LANP= ALLOW |DEFER]     X

               [,DEFLTF=DEFER| FAIL ]    X
               [,DISKF=DEFER| FAIL ]     X
               [,RDRF=DEFER| FAIL ]      X
               [,NODEF=DEFER| FAIL ]     X
               [,CMDF=DEFER| FAIL ]      X
               [,LANF=DEFER| FAIL ]      X

               [,DEFLTW=DEFER|FAIL]      X
```

```
                [,DISKW=DEFER|FAIL]       X
                [,RDRW=DEFER|FAIL]        X
                [,NODEW=DEFER|FAIL]       X
                [,CMDW=DEFER|FAIL]        X
                [,LANW=DEFER|FAIL]        X

                [,DEFLTU=ALLOW|DEFER|FAIL]X
                [,DISKU=ALLOW|DEFER|FAIL] X
                [,RDRU=ALLOW|DEFER|FAIL]  X
                [,NODEU=ALLOW|DEFER|FAIL] X
                [,CMDU=ALLOW|DEFER|FAIL]  X
                [,LANU=ALLOW|DEFER|FAIL]  X

                [,DISKM=ON|OFF]           X
                [,RDRM=ON|OFF]            X
                [,NODEM=ON|OFF]           X
                [,CMDM=ON|OFF]            X
                [,LANM=ON|OFF]
```

The following parameters are supported by the SYSSEC macro.

DEFLTP=, CMDP=, DISKP=, LANP=, RDRP=, NODEP=
Defines the action z/VM will take when RACF has permitted access for commands that are protected within the VMCMD class, or to the profile protecting a minidisk, Guest LAN, virtual reader, RSCS node, or some other resource for which CP has requested an access check. The following parameters options are allowed.

ALLOW
z/VM allows the access.

DEFER
z/VM processes the request as if RACF were not installed. (The request has been deferred to z/VM.)

DEFLTU=, CMDU=, DISKU=, LANU=, RDRU=, NODEU=
Defines the action z/VM will take when a command that is protected within the VMCMD class, or a minidisk, Guest LAN, virtual reader, RSCS node, or some other resource for which CP has requested an access check. The following parameters options are allowed.

ALLOW
z/VM allows the access.

DEFER
z/VM processes the request as if RACF were not installed. (The request has been deferred to z/VM.)

FAIL
z/VM fails the request.

DEFLTF=, CMDF=, DISKF=, LANF=, RDRF=, NODEF=
Defines the action z/VM will take when RACF has denied access for commands that are protected within the VMCMD class, or to a minidisk, Guest LAN, virtual reader, RSCS node, or some other resource for which CP has requested an access check. The following parameters options are allowed.

DEFER
z/VM processes the request as if RACF were not installed. (The request has been deferred to z/VM.)

FAIL
z/VM fails the request.

DEFLTW=, CMDW=, DISKW=, LANW=, RDRW=, NODEW=
Defines the action z/VM will take when RACF would have denied access for a command that is protected within the VMCMD class, or to a minidisk, Guest LAN, virtual reader, RSCS node, or some other resource for which CP has requested an access check, but the profile for the resource was in WARNING mode.

The following parameters options are allowed.

DEFER
z/VM processes the request as if RACF were not installed. (The request has been deferred to z/VM.)

FAIL
z/VM fails the request.

CMDM=, DISKM=, LANM=, RDRM=, NODEM=
Indicates whether error messages defined by RACF will be displayed at the command issuer's console as well as the error messages issued by z/VM. If an invalid value is specified, the default is for messages to be displayed.

The following parameters options are allowed.

ON
RACF error messages will be issued.

OFF
RACF error messages will be suppressed.

Note: RACF does not generate messages for the generalized resource checking interface (ACIRSCHK).

Example 1

This example shows how SYSSEC is coded in the version of HCPRWA that is shipped with RACF. The access decision from the RACF service machine is not changed as a result of these SYSSEC settings, and all messages are displayed at the command issuer's console. The DEFLTW setting defers any checks that were in warning mode in RACF to native CP authorization.

```
SYSSEC DISKP=ALLOW,DISKU=DEFER,DISKF=FAIL,                    X
       DISKW=DEFER,DISKM=ON,                                  X
       RDRP=ALLOW,RDRU=DEFER,RDRF=FAIL,                       X
       RDRW=DEFER,RDRM=ON,                                    X
       NODEP=ALLOW,NODEU=DEFER,NODEF=FAIL,                    X
       NODEW=DEFER,NODEM=ON,                                  X
       CMDP=ALLOW,CMDU=DEFER,CMDF=FAIL,                       X
       CMDW=DEFER,CMDM=ON,                                    X
       LANP=ALLOW,LANU=DEFER,LANF=FAIL,LANW=DEFER,LANM=ON,    X
       DEFLTP=ALLOW,DEFLTU=DEFER,DEFLTF=FAIL,                 X
       DEFLTW=DEFER
```

Example 2

To fail access where the RACF service machine defers due to access warning mode, code SYSSEC as follows. Additionally, RACF messages for profiles defined in the VMCMD class will be suppressed.

```
SYSSEC DISKP=ALLOW,DISKU=DEFER,DISKF=FAIL,                    X
       DISKW=FAIL,DISKM=ON,                                   X
       RDRP=ALLOW,RDRU=DEFER,RDRF=FAIL,                       X
       RDRW=FAIL,RDRM=ON,                                     X
       NODEP=ALLOW,NODEU=DEFER,NODEF=FAIL,                    X
       NODEW=FAIL,NODEM=ON,                                   X
       CMDP=ALLOW,CMDU=DEFER,CMDF=FAIL,                       X
       CMDW=FAIL,CMDM=OFF,                                    X
       LANP=ALLOW,LANU=DEFER,LANF=FAIL,LANW=FAIL,LANM=ON,     X
       DEFLTP=ALLOW,DEFLTU=DEFER,DEFLTF=FAIL,                 X
       DEFLTW=FAIL
```

You could achieve the same result by coding the preceding example as follows, allowing unspecified parameters to assume their default values.

```
SYSSEC DEFLTW=FAIL,DISKW=FAIL,RDRW=FAIL,NODEW=FAIL,CMDW=FAIL,CMDM=OFF,LANW=FAIL
```

Example 3

To fail access when a profile does not exist, code SYSSEC as follows:

```
SYSSEC DISKP=ALLOW,DISKU=FAIL,DISKF=FAIL,                       X
       DISKW=DEFER,DISKM=ON,                                    X
       RDRP=ALLOW,RDRU=FAIL,RDRF=FAIL,                          X
       RDRW=DEFER,RDRM=ON,                                      X
       NODEP=ALLOW,NODEU=FAIL,NODEF=FAIL,                       X
       NODEW=DEFER,NODEM=ON,                                    X
       CMDP=ALLOW,CMDU=FAIL,CMDF=FAIL,                          X
       CMDW=DEFER,CMDM=ON,                                      X
       LANP=ALLOW,LANU=FAIL,LANF=FAIL,LANW=DEFER,LANM=ON,       X
       DEFLTP=ALLOW,DEFLTU=FAIL,DEFLTF=FAIL,                    X
       DEFLTW=DEFER
```

The preceding example can also be coded as follows:

```
SYSSEC DEFLTU=FAIL,DISKU=FAIL,RDRU=FAIL,NODEU=FAIL,CMDU=FAIL,LANU=FAIL
```

Chapter 2. Diagnose X'A0' Subcodes

RACF provides seven subcodes of DIAGNOSE X'A0' that can be used by application programs. The use of six of the subcodes (X'04', X'30', X'34', X'3C', X'50', and X'54') can be protected using RACF profiles.

RACF profile protection will be in effect and privilege class checking will be ignored for these DIAGNOSE X'A0' subcodes only if all of the following conditions are met:

- Control for DIAG0A0 is turned on
- A VMCMD class profile has been defined for the subcode being issued
- The VMCMD class is active.

If any one or more of the above conditions is not met, privilege class checking is enforced.

The DIAGNOSE X'A0' codes that can be used by application programs are as follows:

X'04'
Verify a user ID and validate the user's password.

X'30'
Query a user's current security label.

X'34'
Update the human-readable-label to security label correlation table (“HR table”) in CP.

X'38'
Obtain the size of, or a copy of the human-readable-label to security label correlation table in CP (“HR table”).

X'3C'
Verify a user ID and validate the user's password phrase.

X'50'
Retrieve RACF configuration information.

X'54'
Generate a PassTicket.

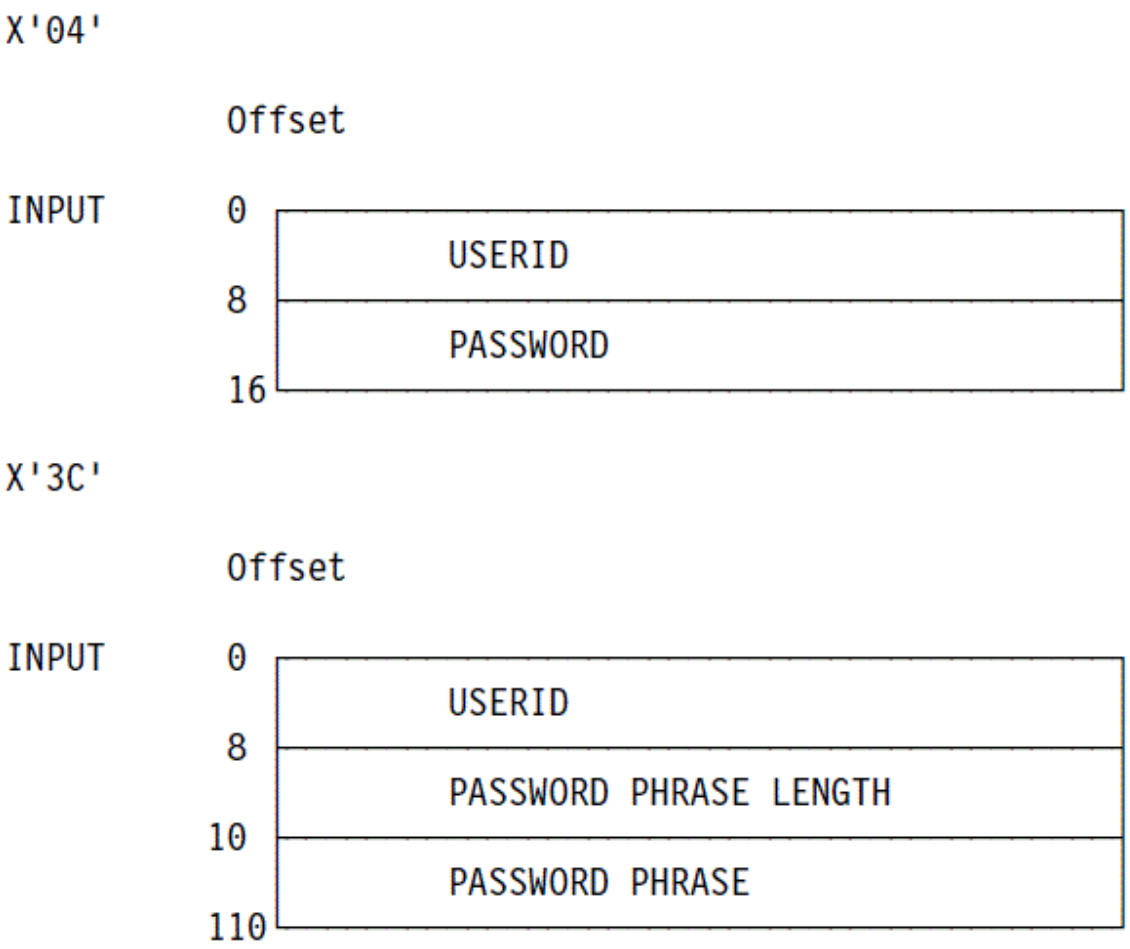
Subcode X'04', Subcode X'3C'

Use subcode X'04' to verify a user and validate the user's password. Use subcode X'3C' to verify a user and validate the user's password phrase.

Authority Required

- If RACF profile protection is in effect, at least READ access to the `DIAG0A0.VALIDATE` resource in the VMCMD class, otherwise
- One or more of **ABCDEF** privilege classes.

Input



RX
The guest real address of a doubleword aligned buffer, formatted as shown above. The user ID and password for subcode X'04' must be padded with blanks.

For X'3C', The guest real address of a 110-byte doubleword aligned buffer:

- The first 8 bytes of this buffer contains a user ID padded with blanks.
- The next 2 bytes contain the password phrase length
- The next 100 bytes (maximum) contain the password phrase of the user ID to be verified. This is a variable length field and need only be as large as the password phrase value.

RY
Subcode value of X'04' to request user verification with password validation or subcode value of X'3C' to request user verification with password phrase validation.

Note: RACF provides RPIVAL MODULE on the RACFVM 305 disk as a command interface to subcodes X'04' and X'3C'. You can invoke this command for the user ID and the password you want to check by using the following syntax:

```
RPIVAL userid password
```

You can invoke this command for the user ID and the password phrase you want to check by using the following syntax:

```
RPIVAL userid 'password phrase'
```

The password phrase can be enclosed within single quotation marks or unquoted. If quoted, embedded single quotation marks within the password phrase value must be doubled.

Output

In most cases, the return code (in RY) will be generated as a result of a RACROUTE REQUEST=VERIFYX macro issued by a RACF service machine.

The other return codes and condition codes are shown in Table 1 on page 21.

Table 1. Condition Codes and Return Codes for Subcode X'04'

Condition Code	Return Code	Status Description
0	RY=X'00'	Password or password phrase correct
1	RY=X'3C'	Password or password phrase not correct
	RY=X'20'	The IUCV path to RACF is broken. The request could not be processed by a RACF service machine.
	All Others	These return code descriptions can be found in *z/VM: Security Server RACROUTE Macro Reference*. See the RACF return codes associated with a SAF return code of X'08' under RACROUTE REQUEST=VERIFYX.
2	RY=X'04'	RACF is inactive (by the SETRACF INACTIVE command). The request could not be processed by a RACF service machine.

Exceptions

specification
If one of the following occurs:

- The guest buffer is not on a doubleword boundary.
- A user ID that is not valid was supplied.
- The RX and RY registers are the same.

privilege
If one of the following occurs:

- The issuer does not have A-F privilege.
- RACF profile protection is in effect, and the issuer is not permitted with at least READ access to the resource named `DIAG0A0.VALIDATE` in the VMCMD class.

Subcode X'30'

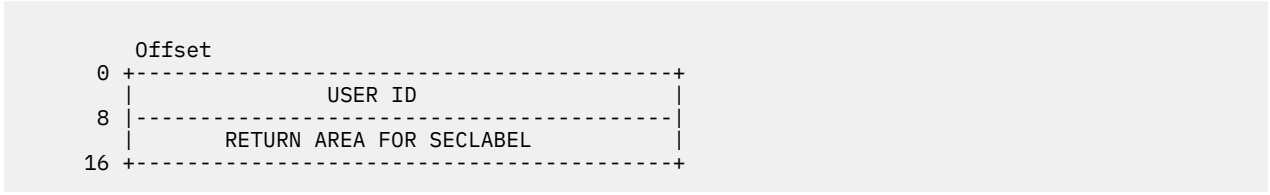
Use subcode X'30' to query the current security label of a user ID.

Authority Required

- If RACF profile protection is in effect, access to the `DIAG0A0.QUERYSEC` profile in the VMCMD class, otherwise
- Privilege class **AB**

Note: No privilege class or VMCMD profile authorization is required to query the security label of your own user ID.

Input



RX
The guest real address of a 16-byte doubleword aligned buffer. The first 8 bytes of this buffer contains a user ID passed as input. The second 8 bytes are used to pass back the security label at which the user is currently logged on.

RY
Subcode value of X'30' to query a security label.

Output

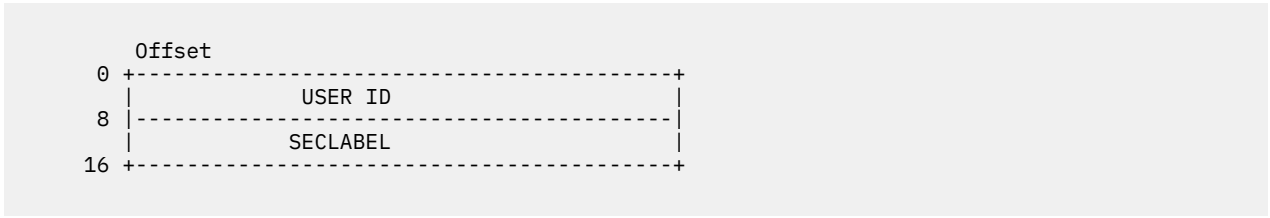


Table 2. Condition Codes and Return Codes for Subcode X'30'

Condition Code	Return Code	Status Description
0	RY = X'00'	The security label for the user ID has been stored in the command issuer's buffer
1	RY = X'00'	The user ID for the security label that is being queried is not logged on or is not logged on with a security label.
3	RY = X'00'	RACF cannot process the request because it is either inactive (by the SETRACF INACTIVE command) or the IUCV path between CP and RACF is broken. No security label is returned.
	RY = X'10'	Processing failed due to a host paging or storage error.

Exceptions

specification
The buffer is not on a doubleword boundary or the RX and RY registers are the same.

privilege
The user is querying someone else's security label but does not have AB privilege or the user is not permitted with at least READ access to the profile named `DIAG0A0.QUERYSEC` in the VMCMD class.

Subcode X'34'

Use subcode X'34' to update the human-readable-label to security label correlation table (“HR table”) in CP.

Authority Required

- If RACF profile protection is in effect, access to the `DIAG0A0.HRTSTORE` profile in the VMCMD class, otherwise

- One or more of **AB** privilege classes.

Input

RX
The address of the HR table.

RX+1
The length in bytes of the HR table.

RY
Subcode value of X'34' to store the HR table in CP.

The HR table consists of contiguous entries in the following format:

```
+------------------------------------ -----------+
|    SECLABLE    |  LEN  |  human readable label  |
+-------------------------------------------------+
```

SECLABEL
An 8-character security label value, padded with blanks.

LEN
The 1-byte length (in bytes) of the subsequent text.

Human Readable Label Label
The descriptive text (not more than 132 characters) associated with the preceding security label.

Output

Table 3. Condition Codes and Return Codes for Subcode X'34'

Condition Code	Return Code	Status Description
0		CP has created a copy of the HR table pointed to by RX.
1		Table length is not valid (must be greater than zero but not greater than 64K)
2		An entry in the table is not valid: • The length specified for the human readable label is less than 1 or greater than 132. • The length specified for the human readable label would extend the entry past the input table length. • An entry is too short to be a valid entry because it is not long enough to include a length-of-label field. Note that this condition can only be detected for the last (as determined by the input table length) entry in the table. **Note:** The RX+1 register will contain the guest real address of the beginning of the entry that is not valid.
3	RY = 0	RACF cannot process the request because it is either inactive (by the SETRACF INACTIVE command) or the IUCV path between CP and RACF is broken.
	RY = X'10'	CP was not able to build the table due to a host paging or storage error.

Exceptions

privilege
RACF profile protection is in effect, and the issuer is not permitted with at least READ access to the profile named `DIAG0A0.HRTSTORE` in the VMCMD class, or the issuer does not have AB privilege.

specification
The register specified as RX was R15.

program
Processing failed while reading from the guest buffer.

Subcode X'38'

Use subcode to X'38' obtain the size of, or a copy of the human-readable-label to security label correlation table (“HR table”).

Authority Required

Privilege class **ANY**

Input

RY
The subcode value of X'38' to obtain the size of, or a copy of the HR table.

Type 1 call (obtain the size of the table):

RX = 0
RX+1 = 0

Type 2 call (obtain a copy of the table):

RX = The address of the user's buffer in which to return the HR table
RX+1 = The length (in doublewords) of the buffer located by RX

Output

For a type 1 call (obtain the size of the table):

Table 4. Condition Codes and Return Codes for Subcode X'38' (Type 1 Call)

Condition Code	Return Code	Status Description
0		RX contains the length in doublewords of the HR table
1		HR table does not exist
3		RACF cannot process the request because it is inactive (by the SETRACF INACTIVE command).

For a type 2 call (obtain a copy of the table):

Table 5. Condition Codes and Return Codes for Subcode X'38' (Type 2 Call)

Condition Code	Return Code	Status Description
0		A copy of the HR table has been stored in the user's buffer.
1		HR table does not exist

Table 5. Condition Codes and Return Codes for Subcode X'38' (Type 2 Call) (continued)		
Condition Code	**Return Code**	**Status Description**
2		HR table does not fit in the specified buffer. RX+1 contains the size in doublewords by which to increase the buffer.
3	RY = X'00'	RACF cannot process the request because it is inactive (by the SETRACF INACTIVE command).
	RY = X'10'	CP was not able to copy the table due to a host paging or storage error.

Exceptions:

specification
The register specified as RX was R15.

program
Processing failed while writing to the guest buffer.

Subcode X'50'

Use subcode to X'50' to retrieve RACF configuration information.

Authority Required

- If RACF profile protection is in effect, access to the `DIAG0A0.RACONFIG` profile in the VMCMD class, otherwise
- One or more of **AB** privilege classes.

Input

RY
The subcode value of X'50'.

RX
The address of the parameter list:

Subcode X'50' parameter list:

Offset	
0	Target server name
8	Length of output buffer
C	Output buffer address
10	Length of request bitmap
14	Data request bitmap

The requested configuration information may be one or more of the following items:

- HCPRWA
- Array of supported bits and lengths
- Dataset name table Range table
- Class descriptor table
- RCVT
- RCVX

- In-storage templates
- Dynamic Parse table
- RACF Router table
- SAF Vector table
- Prefix areas for installed exits
- SMF CPUID
- SMF current disk
- SMF current disk owner
- SMF data file name

See the RPIDIAGS COPY file shipped with the RACF MACLIB for external data structures and DSECTs.

Note:

1. A request with a buffer length insufficient for requested data returns condition code 1 with buffer length updated to necessary buffer size.
2. If server name is blank or zeros, the requested data may be retrieved from any active server, and the plist is updated with the name of the source server.
3. The function will use the existing audit mechanism, updating a VMXEVENT profile with ADDMEM(DIAG0A0/AUDIT)

Output

Table 6. Condition Codes and Return Codes for Subcode X'50' (Type 1 Call)

Condition Code	Return Code	Status Description
0		Requested data in buffer.
1		Buffer too small.
2		Unable to obtain sufficient contiguous buffer.
3		RACF inactive. (Either no IUCV or SETRACF INACTIVE.)

Note:

1. Output for HCPRWA has an additional header to provide the offset to all the CSECTs within the data module:

```
DC     CL8'HCPRWANM'       Ngroups_max
DC     A(HCPRWANM-HCPRWA)
DC     CL8'HCPRWATB'       RACF server table
DC     A(HCPRWATB-HCPRWA)
DC     CL8'HCPRWAPP'       SSI dataarea
DC     A(HCPRWAPP-HCPRWA)
DC     CL8'HCPRWACL'       Change Log server area
DC     A(HCPRWACL-HCPRWA)
DC     CL8'HCPRPITR'       RC translation table
DC     A(HCPRPITR-HCPRWA)
DC     CL8'HCPRWAGD'       Global mdisk table
DC     A(HCPRWAGD-HCPRWA)
DC     XL8'FFFFFFFFFFFFFFFF'  Fence
```

2. Any of the following exits which are loaded in the RACF server will be included:
   - ICHNCV00
   - ICHRIX01
   - ICHRCX01
   - ICHRDX01

- ICHRIX02
- ICHRCX02
- ICHCNX00
- ICHCCX00
- ICHRFX01
- ICHRLX01
- ICHRLX02
- ICHPWX01
- ICHRDX02
- ICHDEX01
- ICHRFX02
- ICHPWX11
- ICHRTX00

The entry for each included exit has the following format:

```
XITENTRY DSECT
XNAME    DS    CL8       Exit name
XADDR    DS    A         Exit load address
XFULL    DS    F         Length of loaded exit
XLEN     DS    F         Length of following prefix
XPREFIX  DS    C         Up to 256 bytes of exit
```

Exceptions:

specification

- Buffer is not on a doubleword boundary.
- The RX and RY registers are the same.
- BITMAP in the request has an unrecognized bit set.

privilege

- The issuer does not have A-B privilege, or
- Does not have at least READ access to the profile named DIAG0A0.RACONFIG in the VMCMD class.

Subcode X'54'

Use subcode X'54' to generate a PassTicket using a user ID and application name.

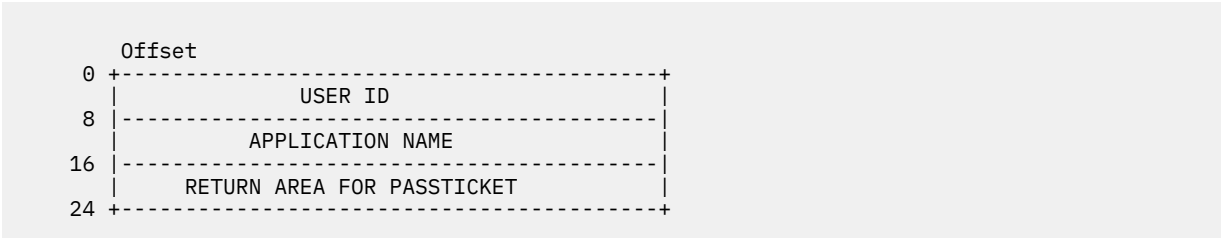
Authority Required

- Update access to the IRRPTAUTH.*application*.*target-userid* profile in the PTKTDATA class, where *application* is the name of a profile in the PTKTDATA class associated with the application to which the *target-userid* wishes to gain access. See "Defining Profiles in the PTKTDATA Class" in "Chapter 19. Using the Secured Signon Function" in *z/VM: RACF Security Server Security Administrator's Guide*.
- Privilege class **ANY**

Input

RX

The guest real address of a 24-byte doubleword aligned buffer. The first 8 bytes of this buffer contains a user ID passed as input. The second 8 bytes contains an application name passed as input. The third 8 bytes are used to pass the generated PassTicket.



RY
Subcode value of X'54' to generate a PassTicket.

Output

Table 7. Condition Codes and Return Codes for Subcode X'54'

Condition Code	Return Code	Status Description
0	RY = X'00'	The PassTicket has been stored in the issuer's buffer.
1	RY = return code	In most cases, the return code is generated as a result of a RACROUTE REQUEST=AUTH issued by a RACF/VM server. These return code descriptions can be found in *z/VM: Security Server RACROUTE Macro Reference*.
3		RACF cannot process the request because it is inactive (by the SETRACF INACTIVE command). No PassTicket is returned.

Exceptions

specification
The buffer is not on a doubleword boundary or the RX and RY registers are the same.

Chapter 3. SMF Records

RACF produces three SMF records:

- **Type 80**—produced during RACF processing
- **Type 81**—produced at the completion of RACF initialization and the SETROPTS command
- **Type 83**—generated by LDAP. See *z/VM: TCP/IP LDAP Administration Guide* for more information.

The first 18 bytes of type 80 and 81 records represent the standard SMF header without subtypes. The first 24 bytes of type 83 records represent the standard SMF header with subtypes. See *System Management Facilities (SMF)* for information about how to use SMF.

For sorting purposes, the RACF report writer reformats SMF records and uses these reformatted records as input to the modules that produce the RACF reports. There are two types of reformatted records - reformatted process records and reformatted status records. If you want to use the RACF report writer exit (ICHRSMFE) to produce additional reports or to add additional record selection criteria, you should familiarize yourself with the layouts of these reformatted records.

It is recommended that you use the RACROUTE system macro and its request types rather than the independent system macros. The new keywords are only supported using RACROUTE. The following table shows the RACROUTE macro request type, the corresponding independent system macro, and the name which refers to either one.

RACROUTE Request Type	Independent RACF System Macro	Referred to As
REQUEST=AUTH	RACHECK	RACHECK request
REQUEST=DEFINE	RACDEF	RACDEF request
REQUEST=EXTRACT	RACXTRT	RACXTRT request
REQUEST=FASTAUTH	FRACHECK	FRACHECK request
REQUEST=LIST	RACLIST	RACLIST request
REQUEST=STAT	RACSTAT	RACSTAT request
REQUEST=VERIFY	RACINIT	RACINIT request
REQUEST=VERIFYX	RACINIT	RACINIT request

Record Type 80: RACF Processing Record

RACF writes record type 80 for the following detected events:

- **Unauthorized attempts to enter the system.** For example, during RACF processing of a RACINIT request macro instruction, RACF found that a RACF-defined user either (1) has supplied an invalid password, password phrase, or group name, (2) is not authorized access to the terminal, or (3) had insufficient security label authority.

  RACF always writes this violation record when it detects the unauthorized attempt; this violation record supplements the information that RACF sends to the security console in RACF message ICH408I.

  Note: The audit record contains a log string indicating what z/VM event was issued.

- **Authorized attempts to enter the system.** RACF provides a RACINIT request option to log successful signons and signoffs as well as ENVIR=CREATE or ENVIR=DELETE signons and signoffs. For the LOG keyword on the RACROUTE and RACINIT request macros, LOG=ALL or LOG=ASIS may be specified to control the generation of log records for RACINIT request. The value of the LOG keyword is passed to

both the RACINIT request preprocessing and postprocessing installation exits. Both exits are invoked prior to the generation of a log record, and the LOG keyword value can be changed for both exits.

Note: The audit record contains a log string indicating what z/VM event was issued.

- **Authorized accesses or unauthorized attempts to access RACF-protected resources.** During RACF processing of a RACHECK or RACDEF request macro instruction, RACF found that one of the following events occurred:

  1. The user was permitted access to a RACF-protected resource and allowed to perform the requested operation.
  2. The user did not have sufficient access or group authority to access a RACF-protected resource, or supplied invalid data while attempting to perform an operation on a RACF-protected resource.

  In the first case, RACF writes the record if the ALL or SUCCESS logging option is set in the resource profile by the ADDSD, ALTDSD, RALTER, RDEFINE, ADDFILE, ADDDIR, ALTFILE, or ALTDIR command and the access type is within the scope of the valid access types. RACF also writes the record if logging has been unconditionally requested by a RACHECK request postprocessing exit routine.

  In the second case, RACF writes the violation record if the ALL or FAILURES logging option is set in the resource profile by the ADDSD, ALTDSD, RALTER, RDEFINE, ADDFILE, ADDDIR, ALTFILE, or ALTDIR command, or if logging is unconditionally requested by a RACHECK request postprocessing exit routine. The violation record supplements the information that RACF sends to the security console in RACF message ICH408I.

  Note that the FAILURES (READ) option is the default in cases where new resources are RACF-protected.

  For the preceding events, a RACHECK request exit routine can modify the logging options by changing the LOG parameter on a RACHECK request macro instruction from ASIS to NOFAIL, NONE, or NOSTAT, or by unconditionally requesting or suppressing logging with the logging control field. (For information on the LOG parameter of a RACHECK request macro instruction, see *z/VM: Security Server RACROUTE Macro Reference*. For information on the logging options of the ADDSD, ALTDSD, ALTUSER, RALTER, RDEFINE, ADDFILE, ADDDIR, ALTFILE, ALTDIR, and SETROPTS commands, see *z/VM: RACF Security Server Command Language Reference*.
- **Authorized or unauthorized attempts to modify profiles on a RACF database.** During RACF command processing, RACF found that a user with the AUDITOR attribute specified that the following be logged:

  1. All detected changes to a RACF database by RACF commands and the RACDEF request
  2. All RACF commands (except LISTDSD, LISTGRP, LISTUSER, RLIST, LDIRECT, LFILE, SRDIR, SRFILE and SEARCH) issued by users with the SPECIAL attribute
  3. All violations detected by RACF commands (except LISTGRP, LISTUSER, RLIST, and SEARCH)
  4. All RACHECK and RACDEF requests issued for the user and all RACF commands (except LISTGRP, LISTUSER, RLIST and SEARCH) issued by the user

  In the first three cases, RACF writes records if a user with the AUDITOR attribute specified AUDIT, SAUDIT, and CMDVIOL, respectively, on the SETROPTS command. In the fourth case, RACF writes the records if a user with the AUDITOR attribute specified UAUDIT on the ALTUSER command.
- **Authorized or unauthorized attempts to issue z/VM events.** z/VM events include CP commands, diagnose codes, certain events related to communication among virtual machines, and certain spool file activities. This auditing is enabled by a VMXEVENT profile. For more information, see "Record Type 80: RACF for z/VM Processing Record for VMXEVENT on z/VM" on page 112 and *z/VM: RACF Security Server Auditor's Guide* (check under "auditing events").

You can use SMF records to:

- Track the total use of a sensitive resource (if the ALL option is set)
- Identify the resources that are repeated targets of detected unauthorized attempts to access them (if the ALL or FAILURES option is set)
- Identify the users who make detected unauthorized requests
- Track SPECIAL user activity

- Track activity of a particular user

In most cases, RACF writes one record for each event. (RACF can write two records for one operation on a resource — for example, when a RACF-protected DASD data set is deleted with scratch.)

SMF record 80 contains the following information:

- The record type
- Time stamp (time and date)
- Processor identification
- Event code and qualifier (explained in Table 1)
- User identification
- Group name
- A count of the relocate sections
- Authorities used to successfully execute commands or access resources
- Reasons for logging
- Command processing error flag
- Foreground user terminal ID
- Foreground user terminal level number
- RACF version, release and modification number
- SECLABEL of user
- The alternate user ID (if any)

(The data in a relocate section is explained in "Table of Relocate Section Variable Data" on page 47 and "Table of Data Type 6 Command-Related Data" on page 64 and "Table of Relocate Section Variable Data for VMXEVENT Class" on page 112.)

The log record RACF creates is a standard type 80 SMF record.

The format of record type 80 is:

Offsets		Name	Length	Format	Description
0	0	SMF80LEN	2	binary	Record length.
2	2	SMF80SEG	2	binary	Segment descriptor.
4	4	SMF80FLG	1	binary	System indicator: **0x00** z/VM All other values indicate z/OS. Use SMF80VRM to determine z/VM release.
5	5	SMF80RTY	1	binary	Record type: 80 (X'50').
6	6	SMF80TME	4	binary	Time of day, in hundredths of a second, that the record was moved to the SMF buffer.
10	A	SMF80DTE	4	packed	Date that the record was moved to the SMF buffer, in the form 0*cyydddF* (where *F* is the sign).
14	E	SMF80SID	4	EBCDIC	System identification (from the SMF CONTROL file).

Offsets		Name	Length	Format	Description
18	12	SMF80DES	2	binary	Descriptor flags **Bit** **Meaning When Set** **0** The event is a violation **1** User is not defined to RACF **2** Record contains a version indicator (see SMF80VER) **3** The event is a warning **4** Record contains a version, release, and modification level number (see SMF80VRM) **5-15** Reserved.
20	14	SMF80EVT	1	binary	Event code.
21	15	SMF80EVQ	1	binary	Event code qualifier.
22	16	SMF80USR	8	EBCDIC	Identifier of the user associated with this event (jobname is used if the user is not defined to RACF).
30	1E	SMF80GRP	8	EBCDIC	Group to which the user was connected (stepname is used if the user is not defined to RACF).
38	26	SMF80REL	2	binary	Offset to the first relocate section from beginning of the record header.
40	28	SMF80CNT	2	binary	Count of the number of relocate sections.
42	2A	SMF80ATH	1	binary	Authorities used for processing commands or accessing resources. (See Note “1” on page 35) **Bit** **Meaning When Set** **0** Normal authority check (resource access) **1** SPECIAL attribute (command processing) **2** OPERATIONS attribute (resource access, command processing) **3** AUDITOR or ROAUDIT attribute (command processing) **4** Installation exit processing (resource access) **5** Failsoft processing (resource access) **6** Bypassed-user ID = *BYPASS* (resource access) **7** Trusted attribute (resource access).

Offsets		Name	Length	Format	Description
43	2B	SMF80REA	1	binary	Reason for logging. These flags indicate the reason RACF produced the SMF record. (See Note “2” on page 35) **Bit** **Meaning When Set** **0** SETROPTS AUDIT(class)—changes to this class of profile are being audited. **1** User being audited **2** SPECIAL users being audited **3** Access to the resource is being audited due to the AUDIT option (specified when profile created or altered by a RACF command), a logging request from the RACHECK exit routine, or because the operator granted access during failsoft processing. **4** RACINIT failure **5** This command is always audited **6** Violation detected in command and CMDVIOL is in effect, or a z/VM event violation is detected **7** Access to entity being audited due to GLOBALAUDIT option.
44	2C	SMF80TLV	1	binary	Terminal level number of foreground user (zero if not available).
45	2D	SMF80ERR	1	binary	Command processing error flag. (See Note “3” on page 36) **Bit** **Meaning When Set** **0** Command had error and RACF could not back out some changes **1** No profile updates were made because of error in RACF processing **2-7** Reserved.
46	2E	SMF80TRM	8	EBCDIC	Terminal ID of foreground user (zero if not available).
54	36	SMF80JBN	8	EBCDIC	Job name. For RACINIT records for batch jobs, this field can be zero.
62	3E	SMF80RST	4	binary	Time, in hundredths of a second, that the reader recognized the JOB statement for this job. For RACINIT records for batch jobs, this field can be zero.
66	42	SMF80RSD	4	packed	Date the reader recognized the JOB statement for this job, in the form 0*cyyddd*F (where *F* is the sign). For RACINIT records for batch jobs, this field can be zero.
70	46	SMF80UID	8	EBCDIC	User identification field from the SMF common exit parameter area. For RACINIT records for batch jobs, this field can be zero. For VMXEVENT audit records, if an alternate user ID is used, the ID is located here.

Offsets		Name	Length	Format	Description
78	4E	SMF80VER	1	binary	Version indicator: **8** RACF/VM 5.4.0 or later. SMF80VRM provides more detail.
79	4F	SMF80RE2	1	binary	Additional reasons for logging **Bit** **Meaning When Set** **0** Security level control for auditing **1** VMEVENT Auditing **2** Class being audited due to SETROPTS LOGOPTIONS **3** Entity audited due to SETROPTS SECLABELAUDIT **4** Entity audited due to SETROPTS COMPATMODE **5** Audited due to SETROPTS COMPATMODE **6** Reserved. **7** Audited because user does not have appropriate authority for OpenExtensions z/VM.
80	50	SMF80VRM	4	EBCDIC	RACF version, release, and modification level. **5040** RACF for z/VM Version 5 Release 4 **6020** RACF for z/VM Version 6 Release 2 **6030** RACF for z/VM Version 6 Release 3 **6040** RACF for z/VM Version 6 Release 4
84	54	SMF80SEC	8	EBCDIC	Security label of the user.
92	5C	SMF80RL2	2	Binary	Offset to extended-length relocate sections.
94	5E	SMF80CT2	2	Binary	Count of extended-length relocate sections.
96	60	SMF80AU2	1	Binary	Authority used continued **Bit** **Meaning When Set** **0** OpenExtensions superuser **1** OpenExtensions system function **2-7** Reserved.
97	61	SMF80RSV	1	Binary	Reserved

Relocate Section:

Offsets		Name	Length	Format	Description
0	0	SMF80DTP	1	binary	Data type.
1	1	SMF80DLN	1	binary	Length of data that follows.
2	2	SMF80DTA	1-255	mixed	Data.

Extended-length Relocate Section:

Offsets		Name	Length	Format	Description
0	0	SMF80TP2	2	Binary	Data type
2	2	SMF80DL2	2	Binary	Length of data that follows
4	4	SMF80DA2	variable	EBCDIC	Data

Note:

1. SMF80ATH: These flags indicate the authority checks made for the user who requested the action. The RACF commands use bits 0, 1, and 3; the RACF requests use bits 0, 2, and 4-7.
   - Bit 0 indicates that the user's authority to issue the command or SVC was determined by the checks for a user with the SPECIAL, OPERATIONS, AUDITOR, or ROAUDIT attribute. This bit indicates that the tests were made, not that the user passed the tests and has authority to issue the command. This bit is not set on if the user has the AUDITOR attribute and entered the command with only those operands that require the AUDITOR attribute.
   - Bit 1 indicates that the user has the SPECIAL attribute and used this authority to issue the command. If the user also has the AUDITOR or ROAUDIT attribute and entered the command with only those operands that require the AUDITOR or ROAUDIT attribute, this bit is not set on because the user did not use his authority as a user with the SPECIAL attribute.
   - Bit 2 is set by the RACHECK and RACDEF requests and indicates that the user has the OPERATIONS attribute and used this authority to obtain access to the resource.
   - Bit 3 indicates that the user has the AUDITOR or ROAUDIT attribute or group-AUDITOR and used this authority to issue the command with operands that require the AUDITOR or ROAUDIT attribute or group-AUDITOR authority.
   - Bit 4 indicates that the user has authority because the exit routine indicated that the request is to be accepted without any further authority checks.
   - Bit 5 indicates that resource access was granted by the operator during failsoft processing.
   - Bit 6 indicates that *BYPASS* was specified on the user ID field. Access was granted because RACF authority checking was bypassed. This bit could also indicate that a violation is detected on a z/VM event.
   - Bit 7 indicates that the user has the trusted attribute.
2. SMF80REA: These flags indicate the reason RACF produced the SMF record.
   - Bit 0 is set when there are changes made to a profile in a class specified in the AUDIT operand of the SETROPTS command.
   - Bit 1 is set when a user with the AUDITOR attribute specifies the UAUDIT operand on the ALTUSER command for a user and the user has changed RACF profiles with a RACF command, or a RACHECK or RACDEF request has been issued for the user.
   - Bit 2 is set when a user with the AUDITOR attribute specifies the SAUDIT operand on the SETROPTS command and a user with the SPECIAL attribute has changed RACF profiles with a RACF command. However, if a user has both the SPECIAL and AUDITOR attributes and issues a command with operands that require only the AUDITOR attribute, RACF does not log this activity because SPECIAL authority was not used.
   - Bit 3 is set if:
     - The AUDIT option in the resource profile specifies that attempts to access the resource be logged.
     - The RACHECK request exit routine specifies unconditional logging.
     - The console operator grants the resource access during failsoft processing.
   - Bit 4 is set when the RACINIT request fails to verify a user because of an invalid group, password, terminal, or OIDCARD.
   - Bit 5 is set if the RVARY or SETROPTS command produced the SMF record. (The execution of these two commands always produce an SMF record.)

- Bit 6 is set when a user with the AUDITOR attribute specifies logging of command violations (with the CMDVIOL operand on the SETROPTS command) and RACF detects a violation.
- Bit 7 is set when attempts to access a RACF-protected resource are being logged, as requested by the GLOBALAUDIT option in the resource profile.

3. SMF80ERR: These flags indicate errors during command processing and the extent of the processing.
   - Bit 0 indicates that an error occurred that prevented the command from completing all updates requested, and the command was unable to back out the updates already done. If this bit is on, there may be an inconsistency between the profiles on the RACF database, or between the profile for a data set and the RACF-indicator for the data set in the DSCB or catalog. The latter is also indicated by a bit in the command-related information for the ADDSD, ALTDSD, and DELDSD commands. For some commands (for example, ADDUSER), the inconsistency means an incompletely defined resource. For other commands, where the profiles are already defined (for example, ALTUSER), the inconsistency means that all changes were not made, but the profiles are still usable.

     This bit indicates a terminating error and should not be confused with a keyword violation or processing error where the command continues processing other operands.
   - Bit 1 indicates that none of the requested changes were made, because either (1) a terminating error occurred before the changes were made, or (2) the command was able to back out the changes after a terminating error.

Table of Event Codes and Event Code Qualifiers

This table describes the SMF80EVT (event code) and SMF80EVQ (event code qualifier) fields.

There are exceptions for event code 1 (logon/logoff): event qualifier codes 8, 12, 13, and 40 are not violations or warnings.

For event codes 8 through 25, an event code qualifier of 1 indicates one of the following:

- The command user is not RACF-defined.
- The command user is not authorized to change the requested profiles on the RACF database.
- The command user does not have sufficient authority for any of the operands on the command.

For event codes 8 through 25, an event code qualifier of 2 indicates that the command user does not have sufficient authority to specify some of the operands, but RACF performed the processing for the operands for which the user has sufficient authority.

Event code qualifiers of 3 and 4 apply to the ADDSD, ALTDSD, and DELDSD commands. They indicate whether the retrieval of the data set affected by the SECLABEL change was successful (3) or not (4).

Note: The event code qualifier is 0 if the recorded event is not a violation or a warning.

For detailed descriptions of the SMF event code qualifiers, refer to Appendix H, “Event Code Qualifiers ,” on page 423.

Event 1(1): LOGON, (X)AUTOLOG, or password validation by REQUEST=VERIFY(X) or DIAGNOSE 0x88

Code Qualifier Dec(Hex)	Description	Relocate type sections (Possible SMF80DTP Values)
0(0)	Successful Initiation	20,46,47,49,53,55, 443
1(1)	Invalid password	
2(2)	Invalid group	
3(3)	Invalid OIDCARD	
4(4)	Invalid terminal/console	
5(5)	Invalid application	
6(6)	Revoked user attempting access	
7(7)	User ID automatically revoked because of excessive password and password phrase attempts	

Event 1(1): LOGON, (X)AUTOLOG, or password validation by REQUEST=VERIFY(X) or DIAGNOSE 0x88 *(continued)*

Code Qualifier Dec(Hex)	Description	Relocate type sections (Possible SMF80DTP Values)
8(8)	Successful termination	
9(9)	Undefined user ID	
10(A)	Insufficient security label authority	
11(B)	Not authorized to security label	
12(C)	Successful RACINIT initiation	
13(D)	Successful RACINIT delete	
14(E)	System now requires more authority	
15(F)	Remote job entry - job not authorized	
16(10)	SURROGAT class is inactive	
17(11)	Submitter is not authorized by user	
18(12)	Submitter not authorized to security label	
19(13)	User is not authorized to job	
20(14)	WARNING - Insufficient security label authority	
21(15)	WARNING - security label missing from user, job or profile	
22(16)	WARNING - not authorized to security label	
23(17)	Security labels not compatible	
24(18)	WARNING - security labels not compatible	
25(19)	Current PASSWORD has expired	
26(1A)	Invalid new PASSWORD	
27(1B)	Verification failed by installation	
28(1C)	Group access has been revoked	
29(1D)	OIDCARD is required	
30(1E)	Network job entry - job not authorized	
31(1F)	Warning - unknown user from trusted node propagated	
32(20)	Successful initiation using PassTicket	
33(21)	Attempted replay of PassTicket	
35(23)	User automatically revoked because of inactivity	
36(24)	Password phrase is not valid	
37(25)	New password phrase is not valid	
38(26)	Current password phrase has expired	
40(28)	SUCCESSM - Successful Multi-Factor Authentication	
41(29)	INVMFA - Failed Multi-Factor Authentication	
42(2A)	MFAUNAVL - Multi-Factor Authentication unavailable	

Event 2(2): RESOURCE ACCESS (detected by RACHECK request and DIRAUTH function, and VMXEVENT auditing)

Code Qualifier Dec(Hex)	Description	Relocate type sections (Possible SMF80DTP Values)
0(0)	Successful access	1, 3, 4, 5, 15, 16, 17, 20, 33, 38, 46, 48, 49, 51, 53, 54, 55 (see Notes 1 and 2)
1(1)	Insufficient authority	
2(2)	Profile not found - RACFIND specified on macro	
3(3)	Access permitted due to warning	
4(4)	Failed due to PROTECTALL	
5(5)	WARNING issued due to PROTECTALL	
6(6)	Insufficient CATEGORY/SECLEVEL	
7(7)	Insufficient security label authority	
8(8)	WARNING - security label missing from job, user, or profile	
9(9)	WARNING - insufficient security label authority	
10(A)	WARNING - Data set not cataloged	
11(B)	Data set not cataloged	
12(C)	Profile not found - required for authority checking	
13(D)	WARNING - insufficient CATEGORY/SECLEVEL	

Note 1: The SMF80DTP value 4 (access authority allowed) can be less than the SMF80DTP value 3 (access authority requested) in two cases:

- When RACF authorizes access to a user who requested access to a database because the user has the OPERATIONS attribute.
- When the RACHECK request exit routine returns a return code of 12, which indicates that the request should be granted.

Note 2: The SMF80DTP value of 16 appears only when the RACHECK request received an old volume (OLDVOL) as input. The value of 33 appears when a generic profile is used.

Event 3(3): ADDVOL/CHGVOL (detected by RACDEF request TYPE=ADDVOL or CHGVOL)

Code Qualifier Dec(Hex)	Description	Relocate type sections (Possible SMF80DTP Values)
0(0)	Successful processing of new volume	1, 4, 5, 15, 16, 17, 33, 38, 44, 46, 49, 53, 55 (see Note)
1(1)	Insufficient authority (DATASET only)	

Note: The SMF80DTP value of 16 appears only when the RACHECK request received an old volume (OLDVOL) as input. The value of 33 appears when a generic profile is used.

Event 4(4): RENAME RESOURCE (detected by RACDEF request TYPE=DEFINE or NEWNAME

Code Qualifier Dec(Hex)	Description	Relocate type sections (Possible SMF80DTP Values)
0(0)	Successful rename	1, 2, 5, 15, 17, 33, 38, 44, 46, 49, 53, 55
1(1)	Invalid group	
2(2)	User not in group	
3(3)	Insufficient authority	
4(4)	Resource name already defined	
5(5)	User not defined to RACF	
6(6)	Resource not protected	
7(7)	WARNING - resource not protected	
8(8)	User in second qualifier is not RACF-defined	

Note: In cases where the RACDEF request is used to rename a resource (SMF80EVT=4), the data type 33 relocate section can hold a generic resource name that is either the old or the new name, or it can hold the generic profile that protects the old or the new name.

Event 5(5): DELETE RESOURCE (detected by RACDEF request TYPE=DELETE or DELETE)

Code Qualifier Dec(Hex)	Description	Relocate type sections (Possible SMF80DTP Values)
0(0)	Successful scratch	1, 5, 15, 17, 33, 38, 44, 46, 49, 53, 55
1(1)	Resource not found	
2(2)	Invalid volume identification (DATASET only)	

Event 6(6): DELETE 1 VOLUME OF MULTIVOLUME RESOURCE (detected by RACDEF request TYPE=DELETE)

Code Qualifier Dec(Hex)	Description	Relocate type sections (Possible SMF80DTP Values)
0(0)	Successful deletion	1, 5, 8, 15, 17, 38, 44, 46, 49, 53, 55

Event 7(7): DEFINE RESOURCE (detected by RACDEF request TYPE=DEFINE)

Code Qualifier Dec(Hex)	Description	Relocate type sections (Possible SMF80DTP Values)
0(0)	Successful definition	1, 5, 15, 17, 18, 19, 33, 38, 40, 44, 46, 49, 53, 55
1(1)	Group undefined	
2(2)	User not in group	
3(3)	Insufficient authority	
4(4)	Resource name already defined	
5(5)	User not defined to RACF	

Event 7(7): DEFINE RESOURCE (detected by RACDEF request TYPE=DEFINE) *(continued)*

Code Qualifier Dec(Hex)	Description	Relocate type sections (Possible SMF80DTP Values)
6(6)	Resource not protected	
7(7)	WARNING - resource not protected	
8(8)	WARNING - security label missing from job, user, or profile	
9(9)	WARNING - insufficient security label authority	
10(A)	User in second qualifier is not RACF-defined	

EVENT Dec(Hex)	Command	Code Qualifier Dec(Hex)	Description	Relocate type sections (Possible SMF80DTP Values)
8(8)	ADDSD	0(0)	No violations detected	6, 7, 10, 13, 33, 38, 40, 44, 49, 50, 51, 53, 55, 62, 63
		1(1)	Insufficient authority (no update to RACF database)	
		2(2)	Keyword violations detected (partial update to RACF database)	
		3(3)	Successful retrieval of data set names affected by a security label change	
		4(4)	Error during retrieval of data set names affected by a security label change	
9(9)	ADDGROUP	0(0)	No violations detected	6, 7, 37, 38, 44, 49, 53, 55, 63
		1(1)	Insufficient authority (no update to RACF database)	
		2(2)	Keyword violations detected (partial update to RACF database)	
		3(3)	Successful retrieval of data set names affected by a SECLABEL change	
		4(4)	Error during retrieval of data set names affected by a SECLABEL change	
10(A)	ADDUSER	0(0)	No violations detected	6, 7, 8, 28, 37, 38, 40, 44, 49, 53, 55, 440
		1(1)	Insufficient authority (no update to RACF database)	
		2(2)	Keyword violations detected (partial update to RACF database)	
		3(3)	Successful retrieval of data set names affected by a SECLABEL change	
		4(4)	Error during retrieval of data set names affected by a SECLABEL change	
11(B)	ALTDSD	0(0)	No violations detected	6, 7, 10, 11, 33, 38, 40, 41, 44, 49, 50, 51, 53, 55, 62, 63
		1(1)	Insufficient authority (no update to RACF database)	
		2(2)	Keyword violations detected (partial update to RACF database)	
		3(3)	Successful retrieval of data set names affected by a SECLABEL change	
		4(4)	Error during retrieval of data set names affected by a SECLABEL change	

EVENT Dec(Hex)	Command	Code Qualifier Dec(Hex)	Description	Relocate type sections (Possible SMF80DTP Values)
12(C)	ALTGROUP	0(0)	No violations detected	6, 7, 37, 38, 44, 49, 53, 55
		1(1)	Insufficient authority (no update to RACF database)	
		2(2)	Keyword violations detected (partial update to RACF database)	
		3(3)	Successful retrieval of data set names affected by a SECLABEL change	
		4(4)	Error during retrieval of data set names affected by a SECLABEL change	
13(D)	ALTUSER	0(0)	No violations detected	6, 7, 8, 28, 37, 38, 40, 41, 44, 49, 53, 55, 440
		1(1)	Insufficient authority (no update to RACF database)	
		2(2)	Keyword violations detected (partial update to RACF database)	
		3(3)	Successful retrieval of data set names affected by a SECLABEL change	
		4(4)	Error during retrieval of data set names affected by a SECLABEL change	
14(E)	CONNECT	0(0)	No violations detected	6, 38, 49, 53, 55
		1(1)	Insufficient authority (no update to RACF database)	
		2(2)	Keyword violations detected (partial update to RACF database)	
		3(3)	Successful retrieval of data set names affected by a SECLABEL change	
		4(4)	Error during retrieval of data set names affected by a SECLABEL change	
15(F)	DELDSD	0(0)	No violations detected	6, 38, 49, 50, 51, 53, 55, 62, 63
		1(1)	Insufficient authority (no update to RACF database)	
		2(2)	Keyword violations detected (partial update to RACF database)	
		3(3)	Successful retrieval of data set names affected by a SECLABEL change	
		4(4)	Error during retrieval of data set names affected by a SECLABEL change	

EVENT Dec(Hex)	Command	Code Qualifier Dec(Hex)	Description	Relocate type sections (Possible SMF80DTP Values)
16(10)	DELGROUP	0(0)	No violations detected	6, 38, 44, 49, 53, 55
		1(1)	Insufficient authority (no update to RACF database)	
		2(2)	Keyword violations detected (partial update to RACF database)	
		3(3)	Successful retrieval of data set names affected by a SECLABEL change	
		4(4)	Error during retrieval of data set names affected by a SECLABEL change	
17(11)	DELUSER	0(0)	No violations detected	6, 38, 44, 49, 53, 55
		1(1)	Insufficient authority (no update to RACF database)	
		2(2)	Keyword violations detected (partial update to RACF database)	
		3(3)	Successful retrieval of data set names affected by a SECLABEL change	
		4(4)	Error during retrieval of data set names affected by a SECLABEL change	
18(12)	PASSWORD	0(0)	No violations detected	6, 38, 44, 49, 53
		1(1)	Insufficient authority (no update to RACF database)	
		2(2)	Keyword violations detected (partial update to RACF database)	
		3(3)	Successful retrieval of data set names affected by a SECLABEL change	
		4(4)	Error during retrieval of data set names affected by a SECLABEL change	
19(13)	PERMIT (including PERMFILE and PERMDIR)	0(0)	No violations detected	6, 9, 12, 13, 14, 17, 26, 38, 39, 49, 53, 55
		1(1)	Insufficient authority (no update to RACF database)	
		2(2)	Keyword violations detected (partial update to RACF database)	
		3(3)	Successful retrieval of data set names affected by a SECLABEL change	
		4(4)	Error during retrieval of data set names affected by a SECLABEL change	

EVENT Dec(Hex)	Command	Code Qualifier Dec(Hex)	Description	Relocate type sections (Possible SMF80DTP Values)
20(14)	RALTER (including ALTFILE and ALTDIR)	0(0)	No violations detected	6, 7, 9, 10, 11, 17, 24, 25, 29, 33, 38, 40, 41, 44, 49, 50, 51, 53, 55
		1(1)	Insufficient authority (no update to RACF database)	
		2(2)	Keyword violations detected (partial update to RACF database)	
		3(3)	Successful retrieval of data set names affected by a SECLABEL change	
		4(4)	Error during retrieval of data set names affected by a SECLABEL change	
21(15)	RDEFINE (including ADDFILE and ADDDIR)	0(0)	No violations detected	6, 7, 9, 17, 24, 29, 33, 38, 40, 44, 49, 50, 51, 53, 55
		1(1)	Insufficient authority (no update to RACF database)	
		2(2)	Keyword violations detected (partial update to RACF database)	
		3(3)	Successful retrieval of data set names affected by a SECLABEL change	
		4(4)	Error during retrieval of data set names affected by a SECLABEL change	
22(16)	RDELETE (including DELFILE and DELDIR)	0(0)	No violations detected	6, 9, 17, 38, 44, 49, 50, 51, 53, 55
		1(1)	Insufficient authority (no update to RACF database)	
		2(2)	Keyword violations detected (partial update to RACF database)	
		3(3)	Successful retrieval of data set names affected by a SECLABEL change	
		4(4)	Error during retrieval of data set names affected by a SECLABEL change	
23(17)	REMOVE	0(0)	No violations detected	6, 17, 38, 49, 53, 55
		1(1)	Insufficient authority (no update to RACF database)	
		2(2)	Keyword violations detected (partial update to RACF database)	
		3(3)	Successful retrieval of data set names affected by a SECLABEL change	
		4(4)	Error during retrieval of data set names affected by a SECLABEL change	

EVENT Dec(Hex)	Command	Code Qualifier Dec(Hex)	Description	Relocate type sections (Possible SMF80DTP Values)
24(18)	SETROPTS	0(0)	No violations detected	6, 21, 22, 23, 27, 32, 34, 35, 36, 42, 43, 44, 45, 49, 53, 55
		1(1)	Insufficient authority (no update to RACF database)	
		2(2)	Keyword violations detected (partial update to RACF database)	
		3(3)	Successful retrieval of data set names affected by a SECLABEL change	
		4(4)	Error during retrieval of data set names affected by a SECLABEL change	
25(19)	RVARY	0(0)	No violations detected	6, 27, 30, 31, 49, 53, 55
		1(1)	Insufficient authority (no update to RACF database)	
		2(2)	Keyword violations detected (partial update to RACF database)	
		3(3)	Successful retrieval of data set names affected by a SECLABEL change	
		4(4)	Error during retrieval of data set names affected by a SECLABEL change	
26(1A)	APPC SESSION ESTABLISHMENT	0(0)	Partner verification was successful	1, 17, 33, 38, 49, 53, 55
		1(1)	Session established without verification	
		2(2)	Local LU key will expire in <= 5 days	
		3(3)	Partner LU access has been revoked	
		4(4)	Partner LU key does not match this LU key	
		5(5)	Session terminated for security reason	
		6(6)	Required SESSION KEY not defined	
		7(7)	Possible security attack by partner LU	
		8(8)	SESSION KEY not defined for partner LU	
		9(9)	SESSION KEY not defined for this LU	
		10(A)	SNA security-related protocol error	
		11(B)	Profile change during verification	
		12(C)	Expired SESSION KEY	
27(1B)	GENERAL	0(0) ⋮ 99(63)	General purpose auditing These qualifiers are installation defined.	17, 46, 49, 53, 55

EVENT Dec(Hex)	Command	Code Qualifier Dec(Hex)	Description	Relocate type sections (Possible SMF80DTP Values)
28(1C)	DIRECTORY SEARCH	0(0)	Access allowed	17, 256, 257, 258, 259, 260, 261, 262, 263, 264, 265, 266, 267, 268, 269, 270, 291, 295, 297, 298, 299, 307, 308, 309, 310
		1(1)	Not authorized to search directory	
29(1D)	CHECK ACCESS TO DIRECTORY	0(0)	Access allowed	17, 256, 257, 258, 259, 260, 261, 262, 263, 264, 265, 266, 267, 268, 269, 270, 297, 298, 299, 307, 308, 309, 310
		1(1)	Caller does not have requested access authority	
30(1E)	CHECK ACCESS TO FILE	0(0)	Access allowed	17, 256, 257, 258, 259, 260, 261, 262, 263, 264, 265, 266, 267, 268, 269, 270, 298, 299, 307, 308, 309, 310
		1(1)	Caller does not have requested access authority	
31(1F)	CHAUDIT	0(0)	File's audit options changed	17, 256, 257, 258, 259, 260, 261, 262, 263, 264, 265, 266, 292, 293, 294, 307, 308, 309, 310
		1(1)	Caller does not have authority to change user audit options of specified file	
		2(2)	Caller does not have authority to change auditor audit options	
33(21)	CHMOD	0(0)	File's mode changed	17, 256, 257, 258, 259, 260, 261, 262, 263, 264, 265, 266, 289, 290, 296, 307, 308, 309, 310
		1(1)	Caller does not have authority to change mode of specified file	
34(22)	CHOWN	0(0)	File's owner or group owner changed	17, 256, 257, 258, 259, 260, 261, 262, 263, 264, 265, 266, 280, 281, 307, 308, 309, 310
		1(1)	Caller does not have authority to change owner or group owner of specified file	
36(24)	EXEC WITH SETUID/SETGID	0(0)	Successful change of UIDs and GIDs	17, 256, 257, 258, 259, 260, 261, 262, 272, 273, 274, 275, 276, 277, 280, 281
		1(1)	Caller does not have access to the appropriate EXEC.U*uid* profile in the VMPOSIX class. This qualifier is only relevant to z/VM. On z/OS, there are no failure cases.	
		2(2)	Caller does not have access to the appropriate EXEC.G*gid* profile in the VMPOSIX class. This qualifier is only relevant to z/VM. On z/OS, there are no failure cases.	
41(29)	LINK	0(0)	New link created	17, 256, 257, 258, 259, 260, 261, 262, 263, 264, 265, 266, 270, 299, 307, 308, 309, 310
		*	Failures logged as directory search or check access event types	
42(2A)	MKDIR	0(0)	Directory successfully created	17, 256, 257, 258, 259, 260, 261, 262, 263, 264, 265, 266, 290, 296, 307, 308, 309, 310
		*	Failures logged as directory search or check access event types	
43(2B)	MKNOD	0(0)	Successful creation of a node	17, 256, 257, 258, 259, 260, 261, 262, 263, 264, 265, 266, 290, 296, 307, 308, 309, 310
		*	Failures logged as directory search or check access event types	

EVENT Dec(Hex)	Command	Code Qualifier Dec(Hex)	Description	Relocate type sections (Possible SMF80DTP Values)
45(2D)	OPEN (NEW FILE)	0(0)	File successfully created	17, 256, 257, 258, 259, 260, 261, 262, 263, 264, 265, 266, 290, 296, 307, 308, 309, 310
		*	Failures logged as directory search or check access event types	
47(2F)	RENAME	0(0)	Rename successful	17, 256, 257, 258, 259, 260, 261, 262, 263, 264, 265, 266, 270, 271, 278, 279, 299, 302, 307, 308, 309, 310, 311, 312, 313, 314
		*	Failures logged as directory search or check access event types	
48(30)	RMDIR	0(0)	Successful RMDIR	17, 256, 257, 258, 259, 260, 261, 262, 263, 264, 265, 266, 307, 308, 309, 310
		*	Failures logged as directory search or check access event types	
49(31)	SETEGID	0(0)	Successful change of effective GID	17, 256, 257, 258, 259, 260, 261, 262, 275, 276, 277, 281
		1(1)	Not authorized to SETEGID	
50(32)	SETEUID	0(0)	Successful change of effective UID	17, 256, 257, 258, 259, 260, 261, 262, 272, 273, 274, 280
		1(1)	Not authorized to SETEUID	
51(33)	SETGID	0(0)	Successful change of GIDs	17, 256, 257, 258, 259, 260, 261, 262, 275, 276, 277, 281
		1(1)	Not authorized to SETGID	
52(34)	SETUID	0(0)	Successful change of UIDs	17, 256, 257, 258, 259, 260, 261, 262, 272, 273, 274, 280
		1(1)	Not authorized to SETUID	
53(35)	SYMLINK	0(0)	Successful SYMLINK	17, 256, 257, 258, 259, 260, 261, 262, 263, 264, 265, 266, 297, 307, 308, 309, 310
		*	Failures logged as directory search or check access event types	
54(36)	UNLINK	0(0)	Successful UNLINK	17, 256, 257, 258, 259, 260, 261, 262, 263, 264, 265, 266, 302, 307, 308, 309, 310
		*	Failures logged as directory search or check access event types	
56(38)	CHECK FILE OWNER	0(0)	User is the owner	17, 256, 257, 258, 259, 260, 261, 262, 263, 264, 265, 266, 307, 308, 309, 310
		1(1)	User is not the owner	

Table of Relocate Section Variable Data

This table describes the variable data elements of the relocate section. These data elements will be present when SMF80CNT has a non-zero value and are located by using the offset in SMF80REL.

Data Type (SMF80DTP) Dec(Hex)	Data Length (SMF80DLN)	Format	Description (SMF80DTA)
1(1)	1-255	EBCDIC	Resource name or old resource name (RACHECK or RACDEF request, or VMXEVENT auditing)
2(2)	1-255	EBCDIC	New resource name (RACDEF request)
3(3)	1	binary	Access authority requested (RACHECK request) (see Note 1)

Data Type (SMF80DTP) Dec(Hex)	Data Length (SMF80DLN)	Format	Description (SMF80DTA)
4(4)	1	binary	Access authority allowed (RACHECK or RACDEF request) (see Note 1)
5(5)	1	binary	Data set level number (00-99)
6(6)	1-255	mixed	RACF command-related data (see Table 3)
7(7)	1-255	EBCDIC	DATA installation-defined data (ADDUSER, ALTUSER, RALTER, RDEFINE, ADDGROUP, ALTGROUP, ADDSD, ALTDSD, ADDFILE, ALTFILE, ADDDIR, ALTDIR)
8(8)	1-20	EBCDIC	NAME user-name (ADDUSER, ALTUSER)
9(9)	1-255	EBCDIC	Resource name (PERMIT, PERMFILE, PERMDIR, RALTER, RDEFINE, RDELETE, ADDFILE, ALTFILE, DELFILE, ADDDIR, ALTDIR, DELDIR)
10(A)	7	EBCDIC	Volume serial (ALTDSD ADDVOL, RALTER ADDVOL, ADDSD VOLUME). When set on, bit 0 of the first byte indicates that the volume was not processed. **Bytes 2-7** Contain the volume serial number.
11(B)	7	EBCDIC	Volume serial (ALTDSD DELVOL, RALTER DELVOL). When set on, bit 0 of the first byte indicates that the volume was not processed. **Bytes 2-7** Contain the volume serial.
12(C)	9-243		1 to 27 ID names (PERMIT, PERMFILE, PERMDIR), each organized as follows:
	1	binary	Processing flags: **Bit** **Meaning When Set** **0** ID ignored because of processing error (see Note 2) **1-7** Reserved
	8	EBCDIC	ID name
13(D)	1-255	EBCDIC	FROM resource name (PERMIT, PERMFILE, PERMDIR, ADDSD, ADDFILE, ADDDIR)
14(E)	12	EBCDIC	VOLUME volume serial (6 bytes) followed by FVOLUME volume serial (6 bytes) (PERMIT)
15(F)	6	EBCDIC	VOLSER volume serial (RACDEF or RACHECK request) (Note that when a RACHECK request receives a DATASET profile as input, the volume serial logged is the first volume serial contained in the profile's list of volume serials.)

Data Type (SMF80DTP) Dec(Hex)	Data Length (SMF80DLN)	Format	Description (SMF80DTA)
16(10)	6	EBCDIC	OLDVOL volume serial (RACDEF or RACHECK request) (Note that when RACHECK request receives a DATASET profile as input, the volume serial logged is the first volume serial contained in the profile's list of volume serials.)
17(11)	1-8	EBCDIC	Class name (RACDEF or RACHECK request, RDEFINE, RALTER, RDELETE, PERMIT, PERMFILE, PERMDIR, ADDFILE, ALTFILE, DELFILE, ADDDIR, ALTDIR, DELDIR, or VMXEVENT auditing)
18(12)	1-255	EBCDIC	MENTITY model resource name (RACDEF request)
19(13)	6	EBCDIC	Volume serial of model resource (RACDEF request)
20(14)	8	EBCDIC	Application name (RACHECK or RACINIT request processed)
21(15)	10	binary	Current class options (set by SETROPTS or RACF initialization): **Bit** **Meaning When Set** **Byte 1** **0** Statistics are in effect **1** Auditing is in effect **2** Protection is in effect **3** Generic profile processing is in effect **4** Generic command processing is in effect **5** Global access checking active **6** Raclist option in effect **7** Genlist option in effect

Data Type (SMF80DTP) Dec(Hex)	Data Length (SMF80DLN)	Format	Description (SMF80DTA)
		EBCDIC	**Bytes 2-9** Class name **Bit** **Meaning When Set** **Byte 10** **0** Reserved **1** ALWAYS **2** NEVER **3** SUCCESSES **4** FAILURES **5** DEFAULT **6-7** Reserved
22(16)	8	EBCDIC	Class name from STATISTICS/NOSTATISTICS keyword (SETROPTS)
23(17)	8	EBCDIC	Class name from AUDIT/NOAUDIT keyword (SETROPTS)
24(18)	2-247	EBCDIC	Resource name from ADDMEM keyword (RDEFINE, RALTER): **Bit** **Meaning When Set** **Byte 1** **0** Resource name not processed **1** Resource name ignored because command user lacked sufficient authority to perform the operation **Bytes 2-247:** Resource name
25(19)	2-247	EBCDIC	Resource name from DELMEM keyword (RALTER). Bit 0 of the first byte, when set on, indicates that the resource name was not processed. **Bytes 2-247** Contain the resource name.
26(1A)	8	EBCDIC	Class name from FCLASS keyword (PERMIT, PERMFILE, PERMDIR)

Data Type (SMF80DTP) Dec(Hex)	Data Length (SMF80DLN)	Format	Description (SMF80DTA)
27(1B)	8	EBCDIC	Class name from CLASSACT/NOCLASSACT keyword (SETROPTS, RVARY)
28(1C)	9	mixed	Class name from CLAUTH/NOCLAUTH keyword (ADDUSER, ALTUSER). Bit 1 of the first byte, when set on, indicates that the class was ignored because the command user did not have sufficient authority to perform the operation. **Bytes 2-9** Contain the class name.
29(1D)	1-255	EBCDIC	Application data (RDEFINE, RALTER, ADDFILE, ALTFILE, ADDDIR, ALTDIR)
30(1E)	12-55	mixed	RACF database status (RVARY, RACF initialization): **Bit** **Meaning When Set** **Byte 1** **0** Database is active **1** Database is backup **2-7** Reserved **Bytes 2-4** Unit name **Bytes 5-10** Volume **Byte 11:** Sequence number **Byte 12:** 1-44 character data set name
31(1F)	1-44	EBCDIC	Data set name from DATASET operand (RVARY)

Data Type (SMF80DTP) Dec(Hex)	Data Length (SMF80DLN)	Format	Description (SMF80DTA)
32(20)	89	mixed	**Byte** **Description** **1** Password interval value **2** Password history value **3** User ID revoke value **4** Password warning level value **5-84** Password syntax rules value **85** User ID inactive interval **86-89** Indicators **Bit** **Meaning** **0** MODEL(GDG) in effect **1** MODEL(USER) in effect **2** MODEL(GROUP) in effect **3** GRPLIST in effect **4-31** Reserved
33(21)	2-255	mixed	Flag bits
34(22)	8	EBCDIC	Class name from GENERIC/NOGENERIC (SETROPTS)
35(23)	8	EBCDIC	Class name from GENCMD/NOGENCMD (SETROPTS)
36(24)	8	EBCDIC	Class name from GLOBAL/NOGLOBAL (SETROPTS)
37(25)	1-44	EBCDIC	Model name
38(26)	8	EBCDIC	User ID or group name that owns the profile (RACHECK request, RACDEF request, and all the RACF commands that produce log records, except SETROPTS and RVARY). During DEFINE operations, this field contains the owner that the profile is defined with; in all other operations, it contains the current owner. Thus, for owner changes, it contains the old owner.
39(27)	9-243		(Releases Prior to 1.9) 1 to 27 program names (PERMIT, PERMFILE, PERMDIR), each organized as follows:

Data Type (SMF80DTP) Dec(Hex)	Data Length (SMF80DLN)	Format	Description (SMF80DTA)
	1	binary	Processing flags: **Bit** **Meaning When Set** **0** program ignored because of processing error (see Note 2) **1-7** Reserved
39(27)	4-255		(With Release 1.9 or later) Variable number of entity names (PERMIT), each organized as follows:
	2	binary	Processing flags: **Bit** **Meaning When Set** **0** Entity ignored because of processing error **1** PROGRAM class entity **2** CONSOLE class entity **3** TERMINAL class entity **4** JESINPUT class entity **5-15** Reserved
	1	binary	Entity length
	1-39	EBCDIC	Entity name
40(28)	2-45		Category name (ADDSD, ALTDSD, ADDUSER, ALTUSER, RDEFINE, RALTER, ADDFILE, ALTFILE, ADDDIR, ALTDIR, commands and RACDEF request), to be added to the profile, and organized as follows:
		binary	Processing flags: **Bit** **Meaning When Set** **Byte 1** **0** Category name ignored because of processing error **1-7** Reserved
		EBCDIC	Bytes 1-44 (at offset 1): Category name added

Data Type (SMF80DTP) Dec(Hex)	Data Length (SMF80DLN)	Format	Description (SMF80DTA)
41(29)	2-45		Category name (ALTDSD, ALTUSER, ALTFILE, ALTDIR, and RALTER commands) to be deleted from the profile and organized as follows:
		binary	Processing flags: **Bit** **Meaning When Set** **Byte 1** **0** Category name ignored because of processing error **1-7** Reserved
		EBCDIC	Bytes 1-44 (at offset 1): Category name deleted
42(2A)	8	EBCDIC	Class name from SETROPTS RACLIST/NORACLIST
43(2B)	8	EBCDIC	Class name from SETROPTS GENLIST/NOGENLIST
44(2C)	1-255	mixed	Any segment data, except BASE: **Bit** **Meaning When Set** **Byte 1** **0** Keyword ignored because of insufficient authority **1** Delete the segment **2-7** Reserved **Byte 2-9:** Name of segment **Byte 10:** Length of subkeyword **Variable length** The subkeyword specified **Variable length** The value associated with the subkeyword (limited to 245 minus length of subkeyword)
45(2D)	9	EBCDIC	Class and logging options from SETROPTS LOGOPTIONS (mixed format)

Data Type (SMF80DTP) Dec(Hex)	Data Length (SMF80DLN)	Format	Description (SMF80DTA)
			Bytes 1-8 Class name **Bit** **Meaning When Set** **Byte 9** **0** Reserved **1** ALWAYS **2** NEVER **3** SUCCESSES **4** FAILURES **5** DEFAULTS **6-7** Reserved
46(2E)	1-255	EBCDIC	Variable length string of data specified on LOGSTR= keyword on RACROUTE macro
48(30)	8	EBCDIC	User ID to whom data is directed (RECVR= keyword on RACROUTE macro)
49(31)	1-20	EBCDIC	User name from ACEE
50(32)	8	EBCDIC	SECLABEL name (ADDSD, ALTDSD, ALTUSER, ADDFILE, ALTFILE, ADDDIR, ALTDIR, RDEFINE, and RALTER commands) to be added to the profile
51(33)	8	EBCDIC	SECLABEL name (RACHECK request or VMXEVENT auditing) of the resource or SECLABEL name (ALTDSD, ALTUSER, ALTFILE, ALTDIR, RALTER commands) to be deleted from the profile
53(35)	80	mixed	User security token, see "RUTKN" in *z/VM: Security Server RACROUTE Macro Reference*.
54(36)	80	mixed	Resource security token (RACHECK request), see "RUTKN" in *z/VM: Security Server RACROUTE Macro Reference*
55(37)	8	binary	Key to link audit records together for a user's APPC transaction processing work.
62(3E)	1-44	EBCDIC	Data set name affected by a SECLABEL change (used by SMF Type 83 Records on z/OS)
63(3F)	4	EBCDIC	Link value to connect data sets affected by a SECLABEL change with the RACF command that caused the change

Notes:

1. The access flags are:

   Bit
   Access Authority

   0
   ALTER

   1
   CONTROL

   2
   UPDATE

   3
   READ

   4
   NONE

2. This bit is turned on for each ID in the list (data type 12) and each program entity name in the list (data type 39) that was not processed because of a nonterminating error, such as user IDs (specified on the ID operand of the PERMIT, PERMFILE, or PERMDIR command) that are not defined to RACF. If a terminating error, such as a RACF manager error, occurred while processing an ID or entity, this bit is turned on for all remaining IDs or entities that were not processed.

   For the PERMIT DELETE, PERMFILE DELETE, PERMDIR DELETE command, when no terminating error has occurred, this bit is turned ON only if no entry in the access list was deleted for the ID or entity.

Table of Extended-Length Relocate Section Variable Data

This table describes the variable data elements of the extended-length relocate section. These data elements will be present when SMF80CT2 has a non-zero value and are located by using the offset SMF80RL2.

Data Type (SMF80TP2) Dec(Hex)	Data Length (SMF80DL2)	Format	Audited by Event Code	Description (SMF80DA2)
256(100)	2	Binary	All	Audit function code, indicating the calling service. Refer to the description in Appendix I, “OpenExtensions Audit Function Codes,” on page 439.
257(101)	4	Binary	All	Old real UID
258(102)	4	Binary	All	Old effective UID
259(103)	4	Binary	36,49,50,51,52	Old saved UID
260(104)	4	Binary	All	Old real GID
261(105)	4	Binary	All	Old effective GID
262(106)	4	Binary	36,49,50,51,52	Old saved GID
263(107)	1-1023	EBCDIC	28,29,30,31,33,34,35,41, 42,43,44,45,47,48,53,54, 55,56	Requested pathname (see also data type 299) **Note:** For events 47 (rename) and 41 (link), this is the old pathname.
265(109)	4	Binary	28,29,30,31,33,34,35,41, 42,43,44,45,47,48,53,54, 55,56	File owner UID
266(10A)	4	Binary	28,29,30,31,33,34,35,41, 42,43,44,45,47,48,53,54, 55,56	File owner GID

Data Type (SMF80TP2) Dec(Hex)	Data Length (SMF80DL2)	Format	Audited by Event Code	Description (SMF80DA2)
267(10B)	1	Binary	28,29,30	Requested access **Value** **Meaning** **X'04'** Read access **X'02'** Write access **X'01'** Execute access **X'81'** Directory search access **X'87'** Any access Multiple bits may be set.
268(10C)	1	Binary	28,29,30	Access type (bits used to make access check) **Value** **Meaning When Set** **1** 'owner' bits **2** 'group' bits **3** 'other' bits **4** no bits used
269(10D)	1	Binary	28,29,30	Access allowed **Value** **Meaning** **X'04'** Read access **X'02'** Write access **X'01'** Execute/search Multiple bits may be set.
270(10E)	1-1023	EBCDIC	28,29,30,41,47	Second requested pathname (see also data type 299) **Note:** For events 47 (rename) and 41 (link), this is the new pathname.
272(110)	4	Binary	36,50,52	New real UID
273(111)	4	Binary	36,50,52	New effective UID
274(112)	4	Binary	36,50,52	New saved UID
275(113)	4	Binary	36,49,51	New real GID
276(114)	4	Binary	36,49,51	New effective GID
277(115)	4	Binary	36,49,51	New saved GID
278(116)	4	Binary	47	Owner UID of deleted file
279(117)	4	Binary	47	Owner GID of deleted file
280(118)	4	Binary	34,36,50,52	UID input parameter
281(119)	4	Binary	34,36,49,51	GID input parameter

Data Type (SMF80TP2) Dec(Hex)	Data Length (SMF80DL2)	Format	Audited by Event Code	Description (SMF80DA2)
289(121)	4	Binary	33,35	Old mode **Bit** **Meaning** **0-19** Reserved **20** S_ISGID bit **21** S_ISUID bit **22** S_ISVTX bit **23-25** Owner permission bits (read/write/ execute) **26-28** Group permission bits (read/write/ execute) **29-31** Other permission bits (read/write/ execute)
290(122)	4	Binary	33,35,42,43,45	New mode **Bit** **Meaning** **0-19** Reserved **20** S_ISGID bit **21** S_ISUID bit **22** S_ISVTX bit **23-25** Owner permission bits (read/write/ execute) **26-28** Group permission bits (read/write/ execute) **29-31** Other permission bits (read/write/ execute)

Data Type (SMF80TP2) Dec(Hex)	Data Length (SMF80DL2)	Format	Audited by Event Code	Description (SMF80DA2)
292(124)	4	Binary	31	Requested audit options **Byte** **Meaning** **1** Read access audit options **2** Write access audit options **3** Execute/search audit options **4** Reserved In each byte, the following flags are defined: **Value** **Meaning** **X'00'** Don't audit any access attempts **X'01'** Audit successful accesses **X'02'** Audit failed access attempts **X'03'** Audit both successful and failed access attempts
293(125)	8	Binary	31	Old audit options (user and auditor) **Byte** **Meaning** **1** User read access audit options **2** User write access audit options **3** User execute/search audit options **4** Reserved **5** Auditor read access audit options **6** Auditor write access audit options **7** Auditor execute/search audit options **8** Reserved In each byte, the following flags are defined: **Value** **Meaning** **X'00'** Don't audit any access attempts **X'01'** Audit successful accesses **X'02'** Audit failed access attempts **X'03'** Audit both successful and failed access attempts

Data Type (SMF80TP2) Dec(Hex)	Data Length (SMF80DL2)	Format	Audited by Event Code	Description (SMF80DA2)
294(126)	8	Binary	31	New audit options (user and auditor) **Byte** **Meaning** **1** User read access audit options **2** User write access audit options **3** User execute/search audit options **4** Reserved **5** Auditor read access audit options **6** Auditor write access audit options **7** Auditor execute/search audit options **8** Reserved In each byte, the following flags are defined: **Value** **Meaning** **X'00'** Don't audit any access attempts **X'01'** Audit successful accesses **X'02'** Audit failed access attempts **X'03'** Audit both successful and failed access attempts
296(128)	4	Binary	33,42,43,45	Requested file mode **Bit** **Meaning** **0-19** Reserved **20** S_ISGID bit **21** S_ISUID bit **22** S_ISVTX bit **23-25** Owner permission bits (read/write/execute) **26-28** Group permission bits (read/write/execute) **29-31** Other permission bits (read/write/execute)
297(129)	1-1023	EBCDIC	28,29,53	Content of symlink
298(12A)	1-256	EBCDIC	28,29,30	File name being checked

Data Type (SMF80TP2) Dec(Hex)	Data Length (SMF80DL2)	Format	Audited by Event Code	Description (SMF80DA2)
299(12B)	1	Binary	28,29,30, 41,47	Flag indicating whether the requested pathname is the old (or only) pathname or the new pathname. This field is X'01' except for ck_access events where authority to a new name is being checked. The second pathname contains the new name specified. **Value** **Meaning** **X'01'** Old (or only) pathname **X'02'** New pathname
301(12D)	variable	EBCDIC	9,10,12,13	Command segment data
302(12E)	1	Binary	47,54	Last link deleted flag **Value** **Meaning** **X'00'** Last link was not deleted **X'01'** Last link was deleted.
307(133)	8	EBCDIC	28,29,30,31,33,34,41,42, 43,45,47,48,53,54,56	Filepool name
308(134)	8	EBCDIC	28,29,30,31,33,34,41,42, 43,45,47,48,53,54,56	Filespace name
309(135)	4	Binary	28,29,30,31,33,34,41,42, 43,45,47,48,53,54,56	Inode (file serial number)
310(136)	4	Binary	28,29,30,31,33,34,41,42, 43,45,47,48,53,54,56	SCID (file serial number)
311(137)	8	EBCDIC	47	Second filepool name
312(138)	8	EBCDIC	47	Second filespace name
313(139)	4	Binary	47	Second Inode (file serial number)
314(13A)	4	Binary	47	Second SCID (file serial number)

Data Type (SMF80TP2) Dec(Hex)	Data Length (SMF80DL2)	Format	Audited by Event Code	Description (SMF80DA2)
440(1B8)	8	Binary	10, 13	**Byte 1: MFA subkeyword specified flags** **Bit** — **Meaning when set** **0** — PWFALLBACK specified **1** — NOPWFALLBACK specified **2** — [z/OS only] FACTOR specified **3** — [z/OS only] DELFACTOR specified **4** — [z/OS only] ACTIVE specified **5** — [z/OS only] NOACTIVE specified **6** — [z/OS only] TAGS specified **7** — [z/OS only] DELTAGS specified **Byte 2: MFA subkeyword specified flags** **Bit** — **Meaning when set** **0** — [z/OS only] NOTAGS specified **1** — [z/OS only] ADDPOLICY specified **2** — [z/OS only] DELPOLICY **3-7** — Reserved **Bytes 3-8: Reserved**
441(1B9)	Variable	EBCDIC	13	[z/OS only] MFA factor name
442(1BA)	Variable	EBCDIC	13	[z/OS only] MFA tag entry from the TAGS/DELTAGS keyword. When TAGS is specified, the entry value is the tag name and value separated by a colon (:). When DELTAGS is specified, the entry value is the tag name only.

Data Type (SMF80TP2) Dec(Hex)	Data Length (SMF80DL2)	Format	Audited by Event Code	Description (SMF80DA2)
443(1BB)	Variable	Mixed	1	**Byte 1, authentication information:** **Bit** — **Meaning when set** **0** — [z/OS only] Authentication from VLF **1** — User is enabled for MFA **2** — MFA user allowed to fall back **3** — No MFA decision for MFA user **4** — [z/OS only] IBMMFA requested that RACROUTE REQUEST=VERIFY return the password-expired return code. **5** — [z/OS only] IBM MFA requested that RACROUTE REQUEST=VERIFY return the new-password-invalid return code. **6** — [z/OS only] IBM MFA requested that RACROUTE REQUEST=VERIFY return the password-invalid return code, but not to increment the password revoke count (partial success) **7** — Reserved **Byte 2, authenticator used:** **Bit** — **Meaning when set** **0** — Password evaluated **1** — Password successful **2** — Password phrase evaluated **3** — Password phrase successful **4** — PassTicket evaluated **5** — PassTicket successful **6** — MFA authenticated successful **7** — MFA authenticated unsuccessful **Bytes 3-6** — [z/OS only] Reason for no MFA decision Part 1 **Bytes 7-10** — [z/OS only] Reason for no MFA decision Part 2
444(1BC)	Variable	EBCBIC	13	[z/OS only] MFA policy name entry from the ADDPOLICY/DELPOLICY keyword

Table of Data Type 6 Command-Related Data

- ADDSD
- ADDUSER
- ALTUSER
- ALTDSD
- CONNECT
- PERMIT (including PERMDIR and PERMFILE)
- RALTER (including ALTDIR and ALTFILE)
- RDEFINE (including ADDDIR and ADDFILE)
- RVARY
- SETROPTS

This table describes the RACF command-related data associated with data type 6. The actual format and content of the data depends upon the command being logged. Command-related data will not appear in the SMF record if the command user is not RACF-defined. Some of the commands also omit the command-related data if the user is not authorized for the requested profile on the RACF database.

The table is arranged by event code. In each description, the keyword flags contain one flag for each possible keyword that you can specify (explicitly or by default) on the command. The 'flags for keywords specified' field indicates whether the keyword was specified or defaulted.

The 'flags for keywords ignored because of insufficient authority' indicates whether the keyword was ignored because the user did not have sufficient authority to use the keyword. The event code qualifier (SMF80EVQ), described in Table 1, is set to 1 if the command user does not have sufficient authority for any of the keywords specified or taken as defaults. The event code qualifier is set to 2 if the command user does not have sufficient authority for some (but not all) of the keywords specified or taken as defaults. In the latter case, the command continues processing the authorized operands.

The 'flags for keywords ignored due to error conditions' field indicates individual keywords that were not processed for reasons other than insufficient authority. Not all commands (event codes 8-25) have these flags. The keyword errors are not terminating errors (like the errors indicated in SMF80ERR) and the command continues processing other specified operands. In the event of a terminating error, these flags do not necessarily indicate what processing was done or not done. Any keyword errors occurring before the terminating error are indicated, but the keywords not processed because of a terminating error are not indicated. The bits in SMF80ERR indicate whether or not RACF already made changes to the RACF database before the terminating error and if it backed out the changes successfully.

Other fields in the command-related data field indicate the subfields specified (or defaulted) for keywords. The fields are flags for subfields that are keywords (such as SUCCESS subfield of AUDIT); they are data for subfields such as owner name or group name.

For example, if the owner of the profile for USERA issues the command:

```
ALTUSER USERA ADSP GRPACC SPECIAL OWNER(USERB)
```

and USERB, the requested new owner is not RACF-defined, then the command-related data would appear in the log record as:

```
012C0000 00040000 00080000 00E4E2C5
D9C14040 40000000 00000000 00000000
00000000 000000E4 E2C5D9C2 40404000
00000000
```

The first word indicates the keywords specified. The second word indicates the user does not have sufficient authority to use the SPECIAL keyword. The third word indicates there was an error processing the OWNER keyword. At offset X'0D' is the name of the user profile being altered. At offset X'27' is the name of the owner specified on the command. RACF processed the ADSP and GRPACC keywords.

Note: If you use SMF records to reconstruct a RACF database, passwords and OIDCARDs are not contained in the records and require special handling, and statistics updates are not recorded.

Event Code Dec(Hex)	Command	Data Length	Format	Description
8(8)	**ADDSD**	2	binary	Flags for keywords specified:
		2	binary	Flags for keywords ignored because of insufficient authority: Same format as flags for keywords specified.
		44	EBCDIC	Data set name
		8	EBCDIC	Type (UNIT keyword)
		1	binary	Flags for UACC keyword: **Note:** If this is a non-DFP data set, RACF ignores bit 4 when checking access to data sets. **Bit** **Authority Specified** **0** ALTER **1** CONTROL **2** UPDATE **3** READ **4** EXECUTE **5-6** Reserved **7** NONE
		8	EBCDIC	User ID or group name (OWNER keyword)

Event Code Dec(Hex)	Command	Data Length	Format	Description
		1	binary	Flags for AUDIT keyword: (only one set at a time) **Bit** **Option Specified** **0** ALL **1** SUCCESS **2** FAILURES **3** NONE **4-5** SUCCESS qualifier codes: **'00'** READ **'01'** UPDATE **'10'** CONTROL **'11'** ALTER **6—7** FAILURES qualifier codes: **'00'** READ **'01'** UPDATE **'10'** CONTROL **'11'** ALTER
		1	binary	nn (LEVEL keyword)
		1	binary	Flags for RACF processing: **Bit** **Meaning** **0** Data set profile inconsistent with RACF indicator **1** Generic profile name specified **2** FROM entity is longer than 44 characters — entity is passed in relocate type 13 **3-7** Reserved
		8	EBCDIC	User to be notified when this profile denies access
		2	binary	Flags for keywords specified
		2	binary	Flags for keywords ignored. Same format as flags for keywords specified.
		1	EBCDIC	Reserved
		2	binary	File sequence number

Event Code Dec(Hex)	Command	Data Length	Format	Description
		2	binary	Retention period
		8	EBCDIC	FROM class name
		44	EBCDIC	FROM resource name
		8	EBCDIC	FROM volume serial
		44	EBCDIC	SECLEVEL name
		8	EBCDIC	SECLABEL
9(9)	**ADDGROUP**	1	binary	Flags for keywords specified: **Bit** **Keyword Specified** **0** SUPGROUP **1** OWNER **2** NOTERMUACC **3** TERMUACC **4** DATA **5** MODEL **6-7** Reserved
		1	binary	Flags for keywords ignored because of insufficient authority: Same format as flags for keywords specified.
		8	EBCDIC	Group name
		8	EBCDIC	Superior group name (SUPGROUP keyword)
		8	EBCDIC	User ID or group name (OWNER keyword)
10(A)	**ADDUSER**			* The data for event code 10 is identical to the data for event code 13, with these exceptions.
		4	binary	Flags for keywords specified:
		4	binary	Flags for keywords ignored because of insufficient authority: Same format as flags for keywords specified.
		4	binary	Flags for keywords ignored because of error conditions

Event Code Dec(Hex)	Command	Data Length	Format	Description
		1	binary	Flags for other violations: **Bit** **Violation** ***0** Command invoker does not have CLAUTH attribute of USER **1** Command invoker does not have sufficient authority to group ***2** Command invoker does not have sufficient authority to user profile ***3-7** Reserved
		8	EBCDIC	User ID
		8	EBCDIC	Group-name (DFLTGRP keyword)
		8	EBCDIC	*Group name (GROUP keyword)
		1	binary	Flags for AUTHORITY keyword: **Bit** **Authority specified** **0** JOIN **1** CONNECT **2** CREATE **3** USE **4-7** Reserved
		1	binary	Flags for UACC keyword: **Bit** **Authority Specified** **0** ALTER **1** CONTROL **2** UPDATE **3** READ **4-6** Reserved **7** NONE
		8	EBCDIC	User ID or group name (OWNER keyword)
		2	binary	Flags for classes specified (CLAUTH keyword)

Event Code Dec(Hex)	Command	Data Length	Format	Description
		2	binary	Flags for classes ignored because of insufficient authority: Same format as for flags for classes specified. Note that if all classes specified are ignored because of insufficient authority, then the 'flags for keywords ignored because of insufficient authority' field indicates that CLAUTH was ignored.
		2	binary	Flags for additional keywords specified **Bit** **Option** **Byte 0** **0** SECLEVEL **1** NOSECLEVEL **2** SECLABEL **3** NOSECLABEL **4** NOEXPIRED **5** EXPIRED **6** Reserved **7** Reserved **Byte 1** **0-1** Reserved **2** PHRASE **3** NOPHRASE **4-5** Reserved **6** ROAUDIT **7** NOROAUDIT
		2	binary	Flags for additional keywords ignored (authorization): Same format as for additional keywords specified flags.

Event Code Dec(Hex)	Command	Data Length	Format	Description
		2	binary	Flags for additional keywords ignored because of processing error: **Bit** **Option** **Byte 0** **0** SECLEVEL **1** NOSECLEVEL **2** SECLABEL **3** NOSECLABEL ***4-7** Reserved **Byte 1** **0-4** Reserved **5** ROAUDIT **6** NOROAUDIT **7** Reserved
		3	packed	Logon time (packed); if time is not specified, this field contains binary zeroes; if TIME (ANYTIME) is specified, this field contains 'F0F0F0'.
		3	packed	Logoff time (packed); if time is not specified, this field contains binary zeroes; if TIME (ANYTIME) is specified, this field contains 'F0F0F0'.
		1	binary	Logon day **Bit** **Days the user cannot log on** **0** Sunday **1** Monday **2** Tuesday **3** Wednesday **4** Thursday **5** Friday **6** Saturday **7** Day not specified
		4	EBCDIC	REVOKE date
		4	EBCDIC	RESUME date
		44	EBCDIC	SECLEVEL name

Event Code Dec(Hex)	Command	Data Length	Format	Description
		8	EBCDIC	SECLABEL name
		4	binary	Flags for additional keywords specified: **Bit** **Keyword specified** **Byte 0** **0** MFA **1** NOMFA **2-7** Reserved **Byte 1** **0-7** Reserved **Byte 2** **0-7** Reserved **Byte 3** **0-7** Reserved
		4	binary	Flags for additional keywords failed because of insufficient authority: **Bit** **Keyword specified** **Byte 0** **0** MFA **1** NOMFA **2-7** Reserved **Byte 1** **0-7** Reserved **Byte 2** **0-7** Reserved **Byte 3** **0-7** Reserved

Event Code Dec(Hex)	Command	Data Length	Format	Description
		4	binary	Flags for additional keywords ignored because of error conditions: **Bit** **Keyword specified** **Byte 0** **0** MFA **1** NOMFA **2-7** Reserved **Byte 1** **0-7** Reserved **Byte 2** **0-7** Reserved **Byte 3** **0-7** Reserved
11(B)	**ALTDSD**	2	binary	Flags for keywords specified: **Bit** **Keyword** **Byte 0** **0** OWNER **1** UACC **2** AUDIT **3** LEVEL **4** ADDVOL **5** DELVOL **6** SET **7** NOSET **Byte 1** **0** GLOBALAUDIT **1** VOLUME **2** PASSWORD **3** UNIT **4** ALTVOL **5** DATA **6-7** Reserved

Event Code Dec(Hex)	Command	Data Length	Format	Description
		2	binary	Flags for keywords ignored because of insufficient authority: Same format as flags for keywords specified, except that Byte 1, Bit 2 is reserved.
		2	binary	Flags for keywords ignored because of error conditions: Same format as flags for keywords specified, except that Byte 1, Bit 2 is reserved.
		44	EBCDIC	Data set name
		8	EBCDIC	User ID or group name (OWNER keyword)
		1	binary	Flags for UACC keyword: **Note:** If this is a non-DFP data set, RACF ignores bit 4 when checking access to the data set. **Bit** **Authority Specified** **0** ALTER **1** CONTROL **2** UPDATE **3** READ **4** EXECUTE **5-6** Reserved **7** NONE

Event Code Dec(Hex)	Command	Data Length	Format	Description
		1	binary	Flags for AUDIT keyword: **Bit** **Option Specified** **0** ALL **1** SUCCESS **2** FAILURES **3** NONE **4-5** SUCCESS qualifier codes: **'00'** READ **'01'** UPDATE **'10'** CONTROL **'11'** ALTER **6-7** FAILURES qualifier codes: **'00'** READ **'01'** UPDATE **'10'** CONTROL **'11'** ALTER
		1	binary	nn (LEVEL keyword)
		1	binary	Flags for GLOBALAUDIT keyword: Same format as flags for AUDIT keyword.
		6	EBCDIC	Volume serial ID (VOLUME keyword)
		8	EBCDIC	Unit information
		1	binary	Flags for RACF processing: **Bit** **Meaning** **0** Profile inconsistent with RACF indicator. **1** Generic profile name specified **2-7** Reserved

Event Code Dec(Hex)	Command	Data Length	Format	Description
		2	binary	Additional keywords specified: **Bit** **Keyword** **Byte 0** **0** GENERIC **1** WARNING **2** NOWARNING **3** ERASE **4** NOERASE **5** RETPD **6** NOTIFY **7** NONOTIFY **Byte 1** **0** SECLEVEL **1** ADDCATEGORY **2** DELCATEGORY **3** NOSECLEVEL **4** SECLABEL **5** NOSECLABEL **6-7** Reserved
		2	binary	Flags for keywords ignored because of insufficient authority: Same format as flags for keywords specified.
		2	binary	Flags for keywords ignored because of a processing error: Same format as flags for keywords specified.
		2	binary	Retention period
		8	EBCDIC	User to be notified when access denied.
		44	EBCDIC	SECLEVEL name
		8	EBCDIC	SECLABEL name

Event Code Dec(Hex)	Command	Data Length	Format	Description
12(C)	**ALTGROUP**	1	binary	Flags for keywords specified: **Bit** **Keyword Specified** **0** SUPGROUP **1** OWNER **2** NOTERMUACC **3** TERMUACC **4** DATA **5** MODEL **6-7** Reserved
		1	binary	Flags for keywords ignored because of insufficient authority: Same format as flags for keyword's specified.
		1	binary	Flags for other violations: **Bit** **Violation** **0** Lack of proper authority to old SUPGROUP **1-7** Reserved
		8	EBCDIC	Group name
		8	EBCDIC	Superior group name (SUPGROUP keyword)
		8	EBCDIC	User ID or group name (OWNER keyword)
		1	binary	Flags for keywords ignored because of error conditions: Same format as flags for keywords specified.
13(D)	**ALTUSER**			* The data for event code 13 is identical to the data for event code 10, with these exceptions.

Event Code Dec(Hex)	Command	Data Length	Format	Description
		4	binary	Flags for keywords specified: **Bit** **Keyword** **Byte 0** **0** DFLTGRP ***1** GROUP **2** PASSWORD **3** NOPASSWORD **4** NAME **5** AUTHORITY **6** DATA **7** GRPACC **Byte 1** **0** NOGRPACC **1** UACC **2** ADSP **3** NOADSP **4** OWNER **5** SPECIAL **6** NOSPECIAL **7** OPERATIONS **Byte 2** **0** NOOPERATIONS **1** CLAUTH **2** NOCLAUTH **3** AUDITOR **4** NOAUDITOR **5** OIDCARD **6** NOOIDCARD ***7** REVOKE

Event Code Dec(Hex)	Command	Data Length	Format	Description
				Byte 3 ***0** RESUME ***1** UAUDIT ***2** NOUAUDIT **3** MODEL **4** NOMODEL **5** WHEN **6** ADDCATEGORY **7** DELCATEGORY
		4	binary	Flags for keywords ignored because of insufficient authority: Same format as flags for keywords specified.
		4	binary	Flags for keywords ignored because of error conditions: Same format as flags for keywords specified.
		1	binary	Flags for other violations: **Bit** **Violation** ***0** Command invoker does not have CLAUTH attribute of USER **1** Command invoker does not have sufficient authority to group ***2** Command invoker does not have sufficient authority to user profile **3-7** Reserved
		8	EBCDIC	User ID
		8	EBCDIC	Group name (DFLTGRP keyword)
		8	EBCDIC	*Group name (GROUP keyword)
		1	binary	Flags for AUTHORITY keyword: **Bit** **Authority Specified** **0** JOIN **1** CONNECT **2** CREATE **3** USE **4-7** Reserved

Event Code Dec(Hex)	Command	Data Length	Format	Description
		1	binary	Flags for UACC keyword: **Bit** **Authority Specified** **0** ALTER **1** CONTROL **2** UPDATE **3** READ **4-6** Reserved **7** NONE
		8	EBCDIC	User ID (OWNER keyword)
		2	binary	Flags for classes specified (CLAUTH keywords) **Bit** **Option** **Byte 0** **0-1** Reserved **2** USER **3** Reserved **4** DASDVOL **5** TAPEVOL **6** TERMINAL **7** Reserved **Byte 1** **0-7** Reserved
		2	binary	Flags for classes ignored because of insufficient authority: Same format as flags for classes specified. Note that if all classes specified are ignored because of insufficient authority, then the 'flags for keywords ignored because of insufficient authority' field indicates that CLAUTH or NOCLAUTH was ignored.

Event Code Dec(Hex)	Command	Data Length	Format	Description
		2	binary	Flags for additional keywords specified **Bit** **Option** **Byte 0** **0** SECLEVEL ***1** NOSECLEVEL ***2** SECLABEL ***3** NOSECLABEL ***4** NOEXPIRED ***5** EXPIRED ***6** Reserved ***7** Reserved **Byte 1** **0** NOREVOKE ***1** NORESUME ***2** PHRASE ***3** NOPHRASE ***4** PWCLEAN ***5** PWCONVERT **6** ROAUDIT **7** NOROAUDIT
		2	binary	Flags for additional keywords ignored (authorization): Same format as for additional keywords specified flags.

Event Code Dec(Hex)	Command	Data Length	Format	Description
		2	binary	Flags for additional keywords ignored because of processing error: **Bit** **Option** **Byte 0** **0** SECLEVEL ***1** NOSECLEVEL ***2** SECLABEL ***3** NOSECLABEL **4-7** Reserved **Byte 1** **0** PWCLEAN ***1** PWCONVERT **2-4** Reserved **5** ROAUDIT **6** NOROAUDIT **7** Reserved
		3	packed	Logon time (packed); if time is not specified, this field contains binary zeroes; if TIME(ANYTIME) is specified, this field contains 'F0F0F0'.
		3	packed	Logoff time (packed); if time is not specified, this field contains binary zeroes; if TIME(ANYTIME) is specified, this field contains 'F0F0F0'.
		1	binary	Day(s) the user cannot logon **Bit** **Day Specified** **0** Sunday **1** Monday **2** Tuesday **3** Wednesday **4** Thursday **5** Friday **6** Saturday **7** Day not specified
		4	EBCDIC	REVOKE date

Event Code Dec(Hex)	Command	Data Length	Format	Description
		4	EBCDIC	RESUME date
		44	EBCDIC	SECLEVEL name
		8	EBCDIC	SECLABEL name
		4	binary	Flags for additional keywords specified: **Bit** **Keyword specified** **Byte 0** **0** MFA **1** NOMFA **2-7** Reserved **Byte 1** **0-7** Reserved **Byte 2** **0-7** Reserved **Byte 3** **0-7** Reserved
		4	binary	Flags for additional keywords failed because of insufficient authority: **Bit** **Keyword specified** **Byte 0** **0** MFA **1** NOMFA **2-7** Reserved **Byte 1** **0-7** Reserved **Byte 2** **0-7** Reserved **Byte 3** **0-7** Reserved

Event Code Dec(Hex)	Command	Data Length	Format	Description
		4	binary	Flags for additional keywords ignored because of error conditions: **Bit** **Keyword specified** **Byte 0** **0** MFA **1** NOMFA **2-7** Reserved **Byte 1** **0-7** Reserved **Byte 2** **0-7** Reserved **Byte 3** **0-7** Reserved

Event Code Dec(Hex)	Command	Data Length	Format	Description
14(E)	**CONNECT**	2	binary	Flags for keywords specified: **Bit** **Keyword** **Byte 0** **0** GROUP **1** UACC **2** AUTHORITY **3** ADSP **4** NOADSP **5** REVOKE **6** RESUME **7** GRPACC **Byte 1** **0** NOGRPACC **1** OPERATIONS **2** NOOPERATIONS **3** SPECIAL **4** NOSPECIAL **5** AUDITOR **6** NOAUDITOR **7** OWNER
		2	binary	Flags for keywords ignored because of insufficient authority: Same format as flags for keywords specified.
		8	EBCDIC	User ID
		8	EBCDIC	Group name (GROUP keyword)

Event Code Dec(Hex)	Command	Data Length	Format	Description
		1	binary	Flags for UACC keyword: **Bit** **Authority Specified** **0** ALTER **1** CONTROL **2** UPDATE **3** READ **4-6** Reserved **7** NONE
		1	binary	Flags for AUTHORITY keyword: **Bit** **Authority Specified** **0** JOIN **1** CONNECT **2** CREATE **3** USE **4-7** Reserved
		1	binary	Flags for additional keywords specified: **Bit** **Keyword Specified** **0** NOREVOKE **1** NORESUME **2-7** Reserved
		1	binary	Flags for additional keywords ignored because of insufficient authority. Same format as flags for additional keywords specified.
		8	EBCDIC	User ID or group name (OWNER keyword)
		4	packed	REVOKE date, packed
		4	packed	RESUME date,packed

Event Code Dec(Hex)	Command	Data Length	Format	Description
15(F)	**DELDSD**	1	binary	Flags for keywords specified or taken as defaults: **Bit** **Keyword Specified** **0** SET **1** NOSET **2** VOLUME **3** GENERIC **4-7** Reserved
		1	binary	Flags for keywords ignored because of insufficient authority: Same format as flags for keywords specified.
		44	EBCDIC	Data set name
		6	EBCDIC	Volume serial ID (VOLUME keyword)
		1	binary	Flags for RACF processing: **Bit** **Meaning** **0** Profile inconsistent with RACF indicator **1** Generic profile name specified **2-7** Reserved
16(10)	**DELGROUP**	8	EBCDIC	Group name
17(11)	**DELUSER**	8	EBCDIC	User ID
18(12)	**PASSWORD**	1	binary	Flags for keywords specified: **Bit** **Keyword Specified** **0** INTERVAL **1** USER **2** PASSWORD **3** PHRASE **4-7** Reserved
		1	binary	Flags for keywords ignored because of insufficient authority: Same format as flags for keywords specified.
		1	binary	Flags for keywords ignored because of error conditions: Same format as flags for keywords specified.

Event Code Dec(Hex)	Command	Data Length	Format	Description
		4	binary	Change-interval (INTERVAL keyword) **Note:** If the NOINTERVAL keyword is specified, the change-interval changes to 'FF'.
		8	EBCDIC	User ID (USER keyword)
19(13)	**PERMIT (including PERMDIR and PERMFILE)**	2	binary	Flags for keywords specified or taken as defaults: **Bit** **Keyword** **Byte 0** **0** CLASS **1** ID **2** ACCESS **3** FROM **4** DELETE **5** FCLASS **6** VOLUME **7** FVOLUME **Byte 1** **0** GENERIC **1** FGENERIC **2** RESET **3** WHEN **4** RESET(WHEN) **5** RESET(STANDARD) **6-7** Reserved
		2	binary	Flags for keywords ignored because of insufficient authority: Same format as flags for keywords specified, except that bits are not set for RESET(STANDARD) or RESET(WHEN).
		2	binary	Flags for keywords ignored because of error conditions: Same format as flags for keywords specified, except that bits are not set for RESET(STANDARD) or RESET(WHEN).

Event Code Dec(Hex)	Command	Data Length	Format	Description
		2	binary	Flags for CLASS keyword, and for the RESET keyword: **Bit** **Option** **Byte 0** **0-2** Reserved **3** DATASET **4** DASDVOL **5** TAPEVOL **6** TERMINAL **7** Reserved **Byte 1** **0** FROM generic resource **1-5** Reserved **6** Conditional access list is indicated by RESET keyword. **7** Standard access list is indicated by RESET keyword.
		1	binary	Flags for ACCESS keyword: **Note:** If this is a non-DFP data set, RACF ignores bit 4 when checking access to the data set. **Bit** **Authority Specified** **0** ALTER **1** CONTROL **2** UPDATE **3** READ **4** EXECUTE **5-6** Reserved **7** NONE
		2	binary	Flags for FCLASS keyword: Same format as flags for CLASS keyword.

Event Code Dec(Hex)	Command	Data Length	Format	Description
20(14)	**RALTER (including ALTDIR and ALTFILE)**			* The data for event code 20 is identical with the data for event code 21, with these exceptions.
		2	binary	Flags for keywords specified: **Bit** **Keyword** **Byte 0** **0** DATA **1** OWNER **2** UACC **3** LEVEL **4** AUDIT ***5** GLOBALAUDIT ***6** ADDVOL ***7** DELVOL **Byte 1** **0** ADDMEM **1** DELMEM **2** APPLDATA **3** SINGLEDSN ***4** NOSINGLEDSN **5** WARNING **6** NOWARNING **7** WHEN
		2	binary	Flags for keywords ignored because of insufficient authority: Same format as flags for keywords specified.

Event Code Dec(Hex)	Command	Data Length	Format	Description
		2	binary	Flags for class name: **Bit** **Option** **Byte 0** **0-3** Reserved **4** DASDVOL **5** TAPEVOL **6** TERMINAL **7** Reserved **Byte 1** **0** Generic resource name specified **1-7** Reserved
		8	EBCDIC	User ID or group name (OWNER keyword)
		1	binary	Flags for UACC keyword: **Bit** **Authority Specified** **0** ALTER **1** CONTROL **2** UPDATE **3** READ **4** EXECUTE **5-6** Reserved **7** NONE
		1	binary	nn (LEVEL keyword)

Event Code Dec(Hex)	Command	Data Length	Format	Description
		1	binary	Flags for AUDIT keyword: **Bit** **Option Specified** **0** ALL **1** SUCCESS **2** FAILURES **3** NONE **4-5** SUCCESS qualifier codes: **'00'** READ **'01'** UPDATE **'10'** CONTROL **'11'** ALTER **6-7** FAILURES qualifier codes: **'00'** READ **'01'** UPDATE **'10'** CONTROL **'11'** ALTER
		1	binary	*Flags for GLOBALAUDIT keyword: Same format as flags for AUDIT keyword.

Event Code Dec(Hex)	Command	Data Length	Format	Description
		2	binary	Flags for keywords specified: **Bit** **Keyword** **Byte 0** **0** NOTIFY ***1** NONOTIFY **2** TVTOC ***3** NOTVTOC **4** TIMEZONE ***5** NOTIMEZONE **6** ADDCATEGORY ***7** DELCATEGORY **Byte 1** **0** SECLEVEL ***1** NOSECLEVEL **2** FROM **3** FCLASS **4** FVOLUME **5** FGENERIC **6** SECLABEL **7** NOSECLABEL
		2	binary	Flags for keywords ignored because of insufficient authority: Same format as flags for keywords specified.
		8	EBCDIC	USER ID to be notified when profile denies access
		44	EBCDIC	FROM resource name
		6	EBCDIC	FROM volume volser
		8	EBCDIC	FROM class name

Event Code Dec(Hex)	Command	Data Length	Format	Description
		1	binary	LOGON days **Bit** **Day Specified** **0** Sunday **1** Monday **2** Tuesday **3** Wednesday **4** Thursday **5** Friday **6** Saturday **7** No keyword
		3	packed	Logon time, packed. If no subkeyword, then binary zeros.
		3	packed	Logoff time, packed. If no subkeyword, then binary zeros.
		3	packed	TIMEZONE value **Bit** **Value** **0-2** Signed decimal
		44	EBCDIC	SECLEVEL name
		8	EBCDIC	SECLABEL name
21(15)	**RDEFINE (including ADDDIR and ADDFILE)**			* The data for event code 21 is identical to the data for event code 20, with these exceptions.

Event Code Dec(Hex)	Command	Data Length	Format	Description
		2	binary	Flags for keywords specified: **Bit** **Keyword** **Byte 0** **0** DATA **1** OWNER **2** UACC **3** LEVEL **4** AUDIT **5** GLOBALAUDIT **6** ADDVOL **7** DELVOL **Byte 1** **0** ADDMEM **1** DELMEM **2** APPLDATA **3** SINGLEDSN **4** NOSINGLEDSN **5** WARNING **6** NOWARNING **7** WHEN
		2	binary	Flags for keywords ignored because of insufficient authority: Same format as flags for keywords specified.

Event Code Dec(Hex)	Command	Data Length	Format	Description
		2	binary	Flags for class-name **Bit** **Option** **Byte 0** **0-3** Reserved **4** DASDVOL **5** TAPEVOL **6** TERMINAL **7** Reserved **Byte 1** **0** Generic resource name specified **1-7** Reserved
		8	EBCDIC	User ID or group name (OWNER keyword)
		1	binary	Flags for UACC keyword: **Bit** **Authority Specified** **0** ALTER **1** CONTROL **2** UPDATE **3** READ **4** EXECUTE **5-6** Reserved **7** NONE
		1	binary	nn (LEVEL keyword)

Event Code Dec(Hex)	Command	Data Length	Format	Description
		1	binary	Flags for AUDIT keyword: **Bit** **Authority Specified** **0** ALL **1** SUCCESS **2** FAILURES **3** NONE **4-5** SUCCESS qualifier codes: **'00'** READ **'01'** UPDATE **'10'** CONTROL **'11'** ALTER **6-7** FAILURES qualifier codes: **'00'** READ **'01'** UPDATE **'10'** CONTROL **'11'** ALTER
		1	binary	*Reserved

Event Code Dec(Hex)	Command	Data Length	Format	Description
		2	binary	Flags for keywords specified: **Bit** **Option** **Byte 0** **0** NOTIFY ***1** NONOTIFY **2** TVTOC ***3** NOTVTOC **4** TIMEZONE ***5** NOTIMEZONE **6** ADDCATEGORY ***7** DELCATEGORY **Byte 1** **0** SECLEVEL ***1** NOSECLEVEL **2** FROM **3** FCLASS **4** FVOLUME **5** FGENERIC **6** SECLABEL **7** NOSECLABEL
		2	binary	Flags for keywords ignored because of insufficient authority: Same format as flags for keywords specified.
		8	EBCDIC	User ID to be notified when profile denies access
		44	EBCDIC	FROM resource name
		6	EBCDIC	FROM volume volser
		8	EBCDIC	FROM class name

Event Code Dec(Hex)	Command	Data Length	Format	Description
		1	binary	LOGON days **Bit** **Day Specified** **0** Sunday **1** Monday **2** Tuesday **3** Wednesday **4** Thursday **5** Friday **6** Saturday **7** No keyword
		3	packed	Logon time, packed. If no subkeyword, then binary zeros.
		3	packed	Logoff time, packed. If no subkeyword, then binary zeros.
		3	packed	TIMEZONE value **Bit** **Option** **Byte 0** **0-7** Reserved **Byte 1** **0-7** Reserved **Byte 2** **0-3** Reserved **4-7** Time zone
		44	EBCDIC	SECLEVEL name
		8	EBCDIC	SECLABEL name

Event Code Dec(Hex)	Command	Data Length	Format	Description
22(16)	**RDELETE**	2	binary	Flags for class-name **Bit** **Option** **Byte 0** **0-3** Reserved **4** DASDVOL **5** TAPEVOL **6** TERMINAL **7** Reserved **Byte 1** **0** Generic resource name specified **1-7** Reserved
23(17)	**REMOVE**	1	binary	Flags for keywords specified: **Bit** **Keyword Specified** **0** GROUP **1** OWNER **2-7** Reserved
		1	binary	Flags for keywords ignored because of insufficient authority: Same format as flags for keywords specified.
		8	EBCDIC	User ID (to be removed)
		8	EBCDIC	Group name (GROUP keyword)
		8	EBCDIC	User ID or group name (OWNER keyword)

Event Code Dec(Hex)	Command	Data Length	Format	Description
24(18)	**SETROPTS**	3	binary	Flags for keywords specified: **Bit** **Option** **Byte 0** **0** TAPE **1** NOTAPE **2** INITSTATS **3** NOINITSTATS **4** SAUDIT **5** NOSAUDIT **6** STATISTICS **7** NOSTATISTICS **Byte 1** **0** AUDIT **1** NOAUDIT **2** TERMINAL **3** NOTERMINAL **4** INTERVAL (PASSWORD) **5** CMDVIOL **6** NOCMDVIOL **7** DASD **Byte 2** **0** NODASD **1** CLASSACT **2** NOCLASSACT **3** HISTORY or NOHISTORY **4** WARNING or NOWARNING **5** REVOKE or NOREVOKE **6** NORULES or RULEn **7** INACTIVE INTERVAL

Event Code Dec(Hex)	Command	Data Length	Format	Description
		3	binary	Flags for keywords ignored because of insufficient authority: Same format as flags for keywords specified.
		1	binary	Flags for STATISTICS or NOSTATISTICS keyword: **Bit** **Option** **Byte 0** **0-2** Reserved **3** DATASET **4** DASDVOL **5** TAPEVOL **6** TERMINAL **7** Reserved
		1	binary	Flags for keywords ignored: **Bit** **Keyword Specified** **0** MODEL-GDG **1** MODEL-NOGDG **2** MODEL-USER **3** MODEL-NOUSER **4** MODEL-GROUP **5** MODEL-NOGROUP **6** GRPLIST **7** NOGRPLIST

Event Code Dec(Hex)	Command	Data Length	Format	Description
		1	binary	Flags for AUDIT or NOAUDIT keyword: **Bit** **Option Specified** **0** Reserved **1** GROUP **2** USER **3** DATASET **4** DASDVOL **5** TAPEVOL **6** TERMINAL **7** Reserved
		1	binary	Flags for keywords specified: **Bit** **Option Specified** **0** MODEL-GDG **1** MODEL-NOGDG **2** MODEL-USER **3** MODEL-NOUSER **4** MODEL-GROUP **5** MODEL-NOGROUP **6** GRPLIST **7** NOGRPLIST
		1	binary	Change-interval (INTERVAL keyword)
		1	binary	Flags for TERMINAL keyword: **Bit** **Option Specified** **0-2** Reserved **3** READ **4-6** Reserved **7** NONE

Event Code Dec(Hex)	Command	Data Length	Format	Description
		1	binary	Flags for current statistics options after SETROPTS has executed: **Bit** **Option Specified** **0** Reserved **1** Bypass RACINIT statistics **2** Bypass data set statistics **3** Bypass tape volume statistics **4** Bypass DASD volume statistics **5** Bypass terminal statistics **6** Bypass ADSP attribute **7** EGN in effect
		1	binary	Flags for current audit options after SETROPTS has executed: **Bit** **Option Specified** **0** Reserved **1** Log group class **2** Log user class **3** Log data set class **4** Log DASD volume class **5** Log tape volume class **6** Log terminal class **7** Reserved
		1	binary	Reserved

Event Code Dec(Hex)	Command	Data Length	Format	Description
		2	binary	Flags for miscellaneous options after SETROPTS has executed: **Bit** **Option** **Byte 0** **0** Perform terminal authorization checking **1** Terminal UACC=NONE (if this bit is off, terminal UACC=READ) **2** Log RACF command violations **3** Log SPECIAL user activity **5-7** Reserved **Byte 1** **0** Tape volume protection is in effect **1** DASD volume protection is in effect **2** Generic profile processing is in effect for the DATASET class **3** Generic command (GENCMD) processing is in effect for the DATASET class **4** REALDSN is in effect **5** JES-XBMALLRACF is in effect **6** JES-EARLYVERIFY is in effect **7** JES-BATCHALLRACF is in effect
		1	binary	Maximum password interval
		1	binary	Password history generation value
		1	binary	Password revoke value
		1	binary	Password warning level
		80	binary binary EBCDIC	Password syntax rules (eight rules). Each rule has the following basic format: **Byte** **Description** **0** Starting length value **1** Ending length value **2-9** Character content rules for each of the eight possible positions. The character values are: L = Alphanumeric A = Alphabetic N = Numeric V = Vowel C = Consonant W = No vowels c = Mixed consonant m = Mixed numeric v = Mixed vowel $ = National s = Special x = Mixed all * = Anything

Event Code Dec(Hex)	Command	Data Length	Format	Description
		1	binary	User ID inactive interval
		3	binary	Flags for keywords specified: **Bit** **Option** **Byte 0** **0** ADSP **1** NOADSP **2** GENERIC **3** NOGENERIC **4** GENCMD **5** NOGENCMD **6** GLOBAL **7** NOGLOBAL **Byte 1** **0** PREFIX **1** NOPREFIX **2** REALDSN **3** NOREALDSN **4** JES-XBMALLRACF **5** JES-NOXBMALLRACF **6** JES-BATCHALLRACF **7** JES-NOBATCHALLRACF **Byte 2** **0** JES-EARLYVERIFY **1** JES-NOEARLYVERIFY **2** REFRESH **3** PROTECTALL-WARNING **4** PROTECTALL-FAILURE **5** NOPROTECTALL **6** EGN in effect **7** NOEGN in effect

Event Code Dec(Hex)	Command	Data Length	Format	Description
		3	binary	Flags for keywords specified but ignored because of insufficient authority: Same format as flags for keywords specified.
		8	EBCDIC	Single-level data set name prefix
		3	binary	Flags for keywords specified: **Bit** **Keyword** **Byte 0** **0** TAPEDSN **1** NOTAPEDSN **2** NOEOS **3** EOS **4** EOS-SECLEVEL **5** EOS-NOSECLEVEL **6** RETPD **7** WHEN **Byte 1** **0** NOWHEN **1** OPERAUDIT **2** NOOPERAUDIT **3** RVARY SWITCH **4** RVARY ACTIVE/INACTIVE **5** ERASE-ALL **6-7** Reserved **Byte 2** Reserved
		3	binary	Flags for keywords specified but ignored because of insufficient authority: Same format as flags for keywords specified.
		1	binary	Erase on scratch security level
		2	binary	Retention period

Event Code Dec(Hex)	Command	Data Length	Format	Description
		1	binary	Flags for miscellaneous options after SETROPTS processing **Bit** **Option** **Byte 0** **0** PROTECTALL-WARNING **1** PROTECTALL-FAILURES **2** EOS **3** EOS-SECLEVEL **4** TAPEDSN **5** WHEN **6** EOS ALL IN EFFECT (erase everything) **7** Reserved

Event Code Dec(Hex)	Command	Data Length	Format	Description
		5	binary	Miscellaneous options after SETROPTS processing **Bit** **Option** **Byte 0** **0-7** Reserved **Byte 1** **0** GENLIST **1** NOGENLIST **2** RACLIST **3** NORACLIST **4** SECLEVELAUDIT **5** NOSECLEVELAUDIT **6** SECLABELAUDIT **7** NOSECLABELAUDIT **8** SECLABELCONTROL **9** NOSECLABELCONTROL **10** MLQUIET **11** NOMLQUIET **12** MLSTABLE **13** NOMLSTABLE **14** GENERICOWNER **15** NOGENERICOWNER **16** SESSIONINTERVAL **17** NOSESSIONINTERVAL **18** JES NJEUSERID (user ID) **19** JES UNDEFINEDUSER (user ID) **20** COMPATMODE

Event Code Dec(Hex)	Command	Data Length	Format	Description
				Byte 1 **21** NOCOMPATMODE **22** MLS WARNING **23** MLS FAILURES **24** NOMLS **25** MLACTIVE WARNING **26** MLACTIVE FAILURES **27** NOMLACTIVE **28** CATDSNS WARNING **29** CATDSNS FAILURES **30** NOCATDSNS **31** LOGOPTIONS
		4	binary	Flags for keywords specified but ignored because of insufficient authority: Same format as flags for keywords specified.
		1	binary	SECLEVEL audit value (auditing occurs for all resources having at least this value
		2	binary	SESSIONINTERVAL interval
		1	binary	Log options for dataset **Bit** **Keyword Specified** **0** ALWAYS **1** NEVER **2** SUCCESSES **3** FAILURES **4** DEFAULT **5-7** Reserved

Event Code Dec(Hex)	Command	Data Length	Format	Description
		2	binary	Current SETROPTS options for B1 security **Bit** **Keyword Specified** **0** SECLABELAUDIT **1** SECLABELCONTROL **2** MLQUIET **3** MLSTABLE **4** GENERICOWNER **5** COMPATMODE **6** MLS WARNING **7** MLS FAILURES **8** MLACTIVE WARNING **9** MLACTIVE FAILURES **10** CATDSNS WARNING **11** CATDSNS FAILURES **12** APPLAUDIT **13** ADDCREATOR **14-15** Reserved
		8	EBCDIC	User ID for JES NJEUSERID
		8	EBCDIC	User ID for JES UNDEFINEDUSER
		1	binary	Password MINCHANGE interval value
		1	EBCDIC	Reserved

Event Code Dec(Hex)	Command	Data Length	Format	Description
		4	binary	Flags for keywords specified **Bit** **Keyword Specified** **0** Primary language specified **1** Secondary language specified **2** ADDCREATOR specified **3** NOADDCREATOR specified **4-7** Reserved **8** PASSWORD MINCHANGE specified **9** PASSWORD MIXEDCASE specified **10** PASSWORD NOMIXEDCASE specified **11** PASSWORD SPECIALCHARS specified **12** PASSWORD NOSPECIALCHARS specified **13** PASSWORD ALGORITHM specified **14** PASSWORD NOALGORITHM specified **15-31** Reserved
		4	binary	Flags for keywords specified but ignored because of insufficient authority: same format as flags for keywords specified.
		3	EBCDIC	Primary language default
		3	EBCDIC	Secondary language default
		3	EBCDIC	Reserved
		1	binary	Current minimum password change interval (MINCHANGE)
		1	Binary	Current options **Bit** **Option** **0** Mixed case passwords are allowed **1** Special characters are allowed in passwords **2-7** Reserved for IBM use

Event Code Dec(Hex)	Command	Data Length	Format	Description
		1	Binary	Password algorithm in effect **Bit** **Meaning** **0** Existing algorithm as indicated by ICHDEX01 (masking, DES, or installation-defined) **1** KDFAES
25(19)	**RVARY**	1	binary	Flags for keywords specified: **Bit** **Keyword Specified** **0** ACTIVE **1** INACTIVE **2** NOTAPE **3** NOCLASSACT **4** SWITCH **5** DATASET **6** LIST **7** NOLIST
		1	binary	Flags for other violations: **Bit** **Violation** **0** Command denied by operator **1** Nonzero code returned from RACF manager during ACTIVE processing **2-7** Reserved

Record Type 80: RACF for z/VM Processing Record for VMXEVENT on z/VM

RACF for z/VM allows you to format SMF type records for the VMXEVENT class on z/VM.

The SMF record for VMXEVENT contains one fixed length portion of record and three relocate sections of record with variable data.

The format of the variable data elements of the relocate section is described below.

Table of Relocate Section Variable Data for VMXEVENT Class

This table describes the variable data elements of the relocate section.

Data Type (SMF80DTP)	Data Length (SMF80DLN)	Format	Description (SMF80DTA)
1	1-255	EBCDIC	Data
17	8	EBCDIC	VMXEVENT
46	1–255	EBCDIC	Logstring: Included for a command run as a result of an AT command showing the originating system. Data is in the form AT_FROM *sysname*
51	8	EBCDIC	Resource security label
53	80	mixed	User security token. See "RUTKN" in *z/VM: Security Server RACROUTE Macro Reference*.

The events that can be audited are divided into 17 groups. The format of the SMF type 80 record created for each of the events is determined by the group number to which the event belongs.

Note: Unless otherwise stated, all commands are in group 4 and all diagnose codes are in group 12.

RACF Event Name	Group Number
APPCCON	(1)
APPCPWVL	(14)
APPCSEV	(2)
AUTOLOG	(16)
CHANGE	(3,4)
GIVE	(13)
IUCVCON	(1)
IUCVSEV	(2)
LINK	(5)
MAINTCCW	(15)
MDISK	(5)
RSTDSEG	(9)
SDF_OPEN	(3)
SDF_CREATE	(3)
SDF_DELETE	(3)
SNIFFER_MODE	(17)
SPF_OPEN	(3)
SPF_CREATE	(3)
SPF_DELETE	(3)
SPTAPE.D	(6)
SPTAPE.E	(6)
TAG	(7)
TRANSFER.D	(4,8)
TRANSFER.G	(4,8)
UTLPRINT	(3)
XAUTOLOG	(16)

RACF Event Name	Group Number
DIAG064	(9)
DIAG068	(10)
DIAG0E4	(11)

The following describes the contents of the data depending on the event being audited and the group to which that event belongs. (Note that any field prefaced with ACI is contained in the z/VM control block, ACIPARMS.) **Group 1** (**Two SMF type 80 records may be created for this group**)

```
Record 1:
```

Offset Dec(Hex)	Field Content	Length	Format	Description
0(0)	ACIEVENT	12	EBCDIC	C'APPCCON ' or C'IUCVCON '
12(C)	Blank	1	EBCDIC	Blank space
13(D)	ACITUSR	8	EBCDIC	Target user ID
21(15)	Blank	1	EBCDIC	Blank space
22(16)	Text	8	EBCDIC	'PATHID ='
30(1E)	Blank	1	EBCDIC	Blank space
31(1F)	ACIPATH	4	EBCDIC	Pathid number
35(23)	Blank	1	EBCDIC	Blank space if connection is being done on behalf of another user
35(23)	Text	1	EBCDIC	'-' if connection is not being done on behalf of another user, and if a second record exists
36(24)	ACISERVR	8	EBCDIC	User ID of invoker when connection being done on behalf of another user
44(2C)	Text	1	EBCDIC	'-' if connection is being done on behalf of another user, and a second record exists

Record 2 (Only created when target of CONNECT is a private server or gateway)

Offset Dec(Hex)	Field Content	Length	Format	Description
0(0)	ACIEVENT	12	EBCDIC	C'APPCCON ' or C'IUCVCON '
12(C)	Blank	1	EBCDIC	Blank space

Offset Dec(Hex)	Field Content	Length	Format	Description
13(D)	ACIQUAL	8	EBCDIC	User ID of private server or LU qualifier for gateway
21(15)	Blank	1	EBCDIC	Blank space
22(16)	ACITLUN	8	EBCDIC	User ID of private server or target LU name

Group 2

Offset Dec(Hex)	Field Content	Length	Format	Description
0(0)	ACIEVENT	12	EBCDIC	C'IUCVSEV ' or C'APPCSEV '
12(C)	Blank	1	EBCDIC	Blank space
13(D)	Text	9	EBCDIC	C 'PATHID ='
22(16)	ACIPATH	4	EBCDIC	Path ID to sever.

Group 3

Offset Dec(Hex)	Field Content	Length	Format	Description
0(0)	ACIEVENT	12	EBCDIC	C'CHANGE ' or C'SPF_CREATE ' or C'SPF_DELETE ' or C'SPF_OPEN ' or C'SDF_CREATE ' or C'SDF_DELETE ' or C'SDF_OPEN ' or C'UTLPRINT '
12(C)	Blank	1	EBCDIC	Blank space
13(D)	ACIORIG	8	EBCDIC	Spool File Origin ID
21(15)	Blank	1	EBCDIC	Blank space
22(16)	ACISPLID	4	EBCDIC	Spool ID

Offset Dec(Hex)	Field Content	Length	Format	Description
26(1A)	Blank	1	EBCDIC	Blank space
27(1B)	Text	10	EBCDIC	C'SFBSTART= '
37(25)	ACIFSTPG	8	EBCDIC	DASD address of the first page
45(20)	Blank	1	EBCDIC	Blank space
46(2E)	ACINSPLD	8	EBCDIC	New spool file security label, if provided

Group 4

Offset Dec(Hex)	Field Content	Length	Format	Description
0(0)	ACIEVENT	12	EBCDIC	CP command name
12(C)	COMDTYPE	8	EBCDIC	CP command type or null if N/A
20(14)	Blank	1	EBCDIC	Blank space
21(15)	ACIDATA	0 - 240	EBCDIC	CP command buffer or auditable event related data passed from ACIPARMS (see Note 1)

Note: One or more additional SMF type 80 records is created if the length of the ACIDATA is greater than 32 characters.

Group 5

Offset Dec(Hex)	Field Content	Length	Format	Description
0(0)	ACIEVENT	12	EBCDIC	C'LINK ' or C'MDISK '
12(C)	Blank	1	EBCDIC	Blank space

Offset Dec(Hex)	Field Content	Length	Format	Description
13(D)	ACITUSR	8	EBCDIC	Target user ID
21(15)	Blank	1	EBCDIC	Blank space
22(16)	ACIADDR	4	EBCDIC	Minidisk address
26(1A)	Blank	1	EBCDIC	Blank space
27(1B)	Text	2	EBCDIC	C'AS'
29(1D)	Blank	1	EBCDIC	Blank space
30(1E)	ACITADDR	4	EBCDIC	Address assigned to the minidisk
34(22)	Blank	1	EBCDIC	Blank space
35(23)	ACIMODE	2	EBCDIC	Access mode granted
37(25)	Blank	1	EBCDIC	Blank space
38(26)	ACITOD	12	EBCDIC	Time of day only for virtual disk (DEFINE VFB-512 command or Directory statement MDISK V-DISK)

Note: An ACIMODE of “XX” indicates that CP has denied the user access to the minidisk due to an error in the LINK command processing.

```
Group 6 (Two SMF type 80 records are created for this group)
```

```
Record 1:
```

Offset Dec(Hex)	Field Content	Length	Format	Description
0(0)	ACIEVENT	12	EBCDIC	C'SPLLOAD ' or 'SPLDUMP '
12(C)	Blank	1	EBCDIC	Blank space

Offset Dec(Hex)	Field Content	Length	Format	Description
13(D)	ACIORIG	8	EBCDIC	Spool file origin ID
21(15)	Blank	1	EBCDIC	Blank space
22(16)	ACISPLID	4	EBCDIC	Spoolid
26(1A)	Text	1	EBCDIC	Continuation mark "-"

Record 2:

Offset Dec(Hex)	Field Content	Length	Format	Description
0(0)	Text	10	EBCDIC	C'SFBSTART= '
10(A)	Blank	1	EBCDIC	Blank space
11(B)	ACIFSTPG	8	EBCDIC	DASD address of the first page
19(13)	Text	7	EBCDIC	C',TOD = '
26(1A)	ACITOD	12	EBCDIC	Timestamp value for SPTAPE calls

Group 7

Offset Dec(Hex)	Field Content	Length	Format	Description
0(0)	EVENT	12	EBCDIC	C'TAG '
12(C)	Blank	1	EBCDIC	Blank space
13(D)	ACINODE	8	EBCDIC	Resource nodename

Group 8 (Two SMF type 80 records are created for this group)

Record 1:

Offset Dec(Hex)	Field Content	Length	Format	Description
0(0)	ACIEVENT	12	EBCDIC	C'TRANSFER '
12(C)	Blank	1	EBCDIC	Blank space
13(D)	ACIORIG	8	EBCDIC	Spool file origin ID
21(15)	Blank	1	EBCDIC	Blank space
22(16)	ACISPLID	4	EBCDIC	Spool ID
26(1A)	Blank	1	EBCDIC	Blank space
27(1B)	Text	2	EBCDIC	C'TO'
29(1D)	Blank	1	EBCDIC	Blank space
30(1E)	ACITUSR	8	EBCDIC	Target user ID
38(26)	Text	1	EBCDIC	Continuation mark "-"

Record 2:

Offset Dec(Hex)	Field Content	Length	Format	Description
0(0)	ACIEVENT	12	EBCDIC	C'TRANSFER '
12(C)	Blank	1	EBCDIC	Blank space
13(D)	ACISPIDN	4	EBCDIC	New spool ID
17(11)	Blank	1	EBCDIC	Blank space
18(12)	Text	10	EBCDIC	C'SFBSTART= '
28(1C)	ACIFSTPG	8	EBCDIC	DASD address of the first page

Group 9

Offset Dec(Hex)	Field Content	Length	Format	Description
0(0)	ACIEVENT	12	EBCDIC	C'DIAG064 ' or C'RSTDSEG '
12(C)	Blank	1	EBCDIC	Blank space
13(D)	ACIRX + ACIRX1	8	EBCDIC	Segment name
21(15)	ACIRY	4	EBCDIC	Subcode, if provided

Group 10

Offset Dec(Hex)	Field Content	Length	Format	Description
0(0)	ACIEVENT	12	EBCDIC	C'DIAG068 '
12(C)	Blank	1	EBCDIC	Blank space
13(D)	ACIDATA(13:24)	12	EBCDIC	DIAG 68 or SCIF function
25(19)	Blank	1	EBCDIC	Blank space
26(1A)	ACITUS	8	EBCDIC	Target user ID

Where

- SCIF functions are:

  CHECKED
  UNCHECKED
- DIAG068 functions are:
  - SEND
  - SEND/RECEIVE
  - SENDX
  - RECEIVE
  - REPLY
  - IDENTIFY

Group 11

Offset Dec(Hex)	Field Content	Length	Format	Description
0(0)	ACIEVENT	12	EBCDIC	C'DIAG0E4 '
12(C)	Blank	1	EBCDIC	Blank space
13(D)	ACITUSR	8	EBCDIC	Target user ID
21(15)	Blank	1	EBCDIC	Blank space
22(16)	ACIADDR	4	EBCDIC	Resource address
26(1A)	Text	2	EBCDIC	C'AS'
28(1C)	Blank	1	EBCDIC	Blank space
29(1D)	ACITADDR	4	EBCDIC	Target address

Group 12

Offset Dec(Hex)	Field Content	Length	Format	Description
0(0)	ACIEVENT	12	EBCDIC	C'DIAGxxx '
12(C)	Blank	1	EBCDIC	Blank space
13(D)	ACIRX	8	EBCDIC	Contents of the RX register
21(15)	Blank	1	EBCDIC	Blank space
22(16)	ACIRX1	8	EBCDIC	Contents of the RX+1 register
30(1E)	Blank	1	EBCDIC	Blank space
31(1F)	ACIRY	8	EBCDIC	Contents of the RY register
39(27)	Blank	1	EBCDIC	Blank space

Offset Dec(Hex)	Field Content	Length	Format	Description
40(28)	ACIRY1	8	EBCDIC	Contents of the RY+1 register

Group 13

Offset Dec(Hex)	Field Content	Length	Format	Description
0(0)	ACIEVENT	12	EBCDIC	C'GIVERETN '
12(C)	Blank	1	EBCDIC	Blank space
13(D)	ACIDETAD	3	EBCDIC	Returning User's virtual address for the device
16(10)	Blank	1	EBCDIC	Blank space
17(11)	Text	2	EBCDIC	C'TO'
19(13)	Blank	1	EBCDIC	Blank space
20(14)	ACITUSR	8	EBCDIC	Target user ID (user receiving device)
21(15)	Blank	1	EBCDIC	Blank space
22(16)	ACIRECAD	3	EBCDIC	Receiving user's virtual address for the device

Group 14

Offset Dec(Hex)	Field Content	Length	Format	Description
0(0)	ACIEVENT	12	EBCDIC	C'APPCPWVL '

Group 15

Offset Dec(Hex)	Field Content	Length	Format	Description
0(0)	ACIEVENT	12	EBCDIC	C'MAINTCCW '
12(C)	Blank	1	EBCDIC	Blank space
13(D)	ACIVOLSR	6	EBCDIC	Volume Serial
19(13)	Blank	1	EBCDIC	Blank space
20(14)	ACIRDEV	4	EBCDIC	Real device address
24(18)	Blank	1	EBCDIC	Blank space
25(19)	ACISCYL	8	EBCDIC	Starting cylinder
33(21)	Blank	1	EBCDIC	Blank space
34(22)	ACIECYL	8	EBCDIC	Ending cylinder

Group 16

Offset Dec(Hex)	Field Content	Length	Format	Description
0(0)	ACIEVENT	12	EBCDIC	C'AUTOLOG ' or C'XAUTOLOG '
12(C)	Text	1	EBCDIC	C'('
13(D)	ACICMDTP	8	EBCDIC	CP command version
21(15)	Text	1	EBCDIC	C')'
22(16)	Blank	1	EBCDIC	Blank space
23(17)	ACITUSR	8	EBCDIC	Target user ID of the command

Group 17

Record 1

Offset Dec(Hex)	Field Content	Length	Format	Description
0(0)	ACIEVENT	12	EBCDIC	C'SNIFFER_MODE'
12(C)	Blank	1	EBCDIC	Field separator
13(D)	Text	3	EBCDIC	C'ON ' if ACILSON is set or C'OFF' if ACILSOFF is set
16(10)	Blank	1	EBCDIC	Field separator
17(11)	ACITUSR.ACILNID	17	EBCDIC	Switch's base profile name
34(22)	Blank	1	EBCDIC	Field separator
35(23)	Text	5	EBCDIC	C'VDEV='
40(28)	ACITADDR	4	EBCDIC	VDEV
44(2C)	Blank	1	EBCDIC	Field separator
45(2D)	Text	6	EBCDIC	C'VLANS('
51(33)	ACILVIDL/2	4	EBCDIC	Count of VLAN ids in list
55(37)	Text	2	EBCDIC	C')='
57(39)	ACIVLANC	5(0-2000)	EBCDIC	An array of VLANids delimited by blanks. Each array element will consist of four digits, with leading zeroes when necessary. **Note:** Only 39 array elements can fit into a single record. If this limit is exceeded, the 39th element will contain a dash as a continuation delimeter, and the list will continue in a subsequent record.

Record Type 81: RACF Initialization Record

RACF writes record type 81 at the completion of the initialization of RACF. This record contains:

- Record type
- Time stamp (time and date)
- Processor identification
- Name of each RACF database
- Volume identification of each RACF database
- Unit name of the RACF database
- RACF options
- The maximum password interval
- The default installation language codes in effect at IPL time.

The format of record type 81 is:

Offsets		Name	Length	Format	Description
0	0	SMF81LEN	2	binary	Record length.
2	2	SMF81SEG	2	binary	Segment descriptor.
4	4	SMF81FLG	1	binary	System indicator **0x00** z/VM All other values indicate z/OS. Use SMF81VRM to determine z/VM release.
5	5	SMF81RTY	1	binary	Record type: 81 (X'51').
6	6	SMF81TME	4	binary	Time of day, in hundredths of a second, that the record was moved to the SMF buffer.
10	A	SMF81DTE	4	packed	Date that the record was moved to the SMF buffer, in the form 0*cyydddF* (where *F* is the sign).
14	E	SMF81SID	4	EBCDIC	System identification (from the SID parameter).
18	12	SMF81RDS	44	EBCDIC	Data set name of the RACF database for this IPL (blanks if RACF is not active).
62	3E	SMF81RVL	6	EBCDIC	Volume identification of RACF database. If the database is split among several DASD volumes, this field equals the first primary data set. If RACF is not active, this field is blank.
68	44	SMF81RUN	3	EBCDIC	Unit name of RACF database; blanks if RACF is not active.
71	47	SMF81UDS	44	EBCDIC	Data set name of the user attribute data set (UADS) data set for this IPL.
115	73	SMF81UVL	6	EBCDIC	Volume identification of the user attribute data set (UADS) data set.

Offsets		Name	Length	Format	Description
121	79	SMF81OPT	1	binary	Options indicator **Bit** **Meaning When Set** **0** No RACINIT statistics are recorded **1** No DATASET statistics are recorded **2** RACINIT preprocessing exit routine, ICHRIX01, is active **3** RACHECK preprocessing exit routine, ICHRCX01, is active **4** RACDEF preprocessing exit routine, ICHRDX01, is active **5** RACINIT post-processing exit routine, ICHRIN02, is active **6** RACHECK post-processing exit routine, ICHRCX02, is active **7** New password exit routine, ICHPWX01, is active.
122	7A	SMF81OP2	1	binary	Options indicator 2 **Bit** **Meaning When Set** **0** No tape volume statistics are recorded **1** No DASD volume statistics are recorded **2** No terminal statistics are recorded **3** Command exit routine ICHCNX00 is active **4** Command exit routine ICHCCX00 is active **5** ADSP is not active **6** Encryption exit routine, ICHDEX01, is active **7** Naming convention table, ICHNCV00 is present.

Offsets		Name	Length	Format	Description
123	7B	SMF81OP3	1	binary	Options indicator 3 **Bit** **Meaning When Set** **0** Tape volume protection is in effect. **1** No duplicate data set names are to be defined **2** DASD volume protection is in effect **3** Record contains version indicator **4** FRACHECK preprocessing exit routine, ICHRFX01, is active **5** RACLIST pre/postprocessing exit routine, ICHRLX01, is active **6** RACLIST selection exit routine, ICHRLX02, is active **7** RACDEF postprocessing exit routine, ICHRDX02, is active.
124	7C	SMF81AOP	1	binary	Audit options **Bit** **Meaning When Set** **0** User class profile changes are being logged **1** Group class profile changes are being logged **2** Data set class profile changes are being logged **3** Tape volume class profile changes are being logged **4** DASD volume class profile changes are being logged **5** Terminal class profile changes are being logged **6** RACF command violations are being logged **7** SPECIAL user activity is being logged.
125	7D	SMF81AO2	1	binary	Audit options 2 **Bit** **Meaning When Set** **0** Operation user activity **1** Audit by security level is in effect **2-7** Reserved.

Offsets		Name	Length	Format	Description
126	7E	SMF81TMO	1	binary	Terminal verification options indicator **Bit** **Meaning When Set** **0** Terminal authorization checking is in effect **1** Universal access for undefined terminals is NONE; if not set, UACC=READ **2** REALDSN is in effect **3** JES-XBMALLRACF is in effect **4** JES-EARLYVERIFY is in effect **5** JES-BATCHALLRACF is in effect **6** FRACHECK post processing exit routine, ICHRFX02, is active **7** Reserved.
127	7F	SMF81PIV	1	binary	Maximum password interval (0-254).
128	80	SMF81REL	2	binary	Offset to the first relocate section from the beginning of the record header.
130	82	SMF81CNT	2	binary	Number of relocate sections.
132	84	SMF81VER	1	binary	Version indicator (6 = RACF Version 1, Release 7). As of RACF 1.8.1, SMF81VRM is used instead.
133	85	SMF81QL	8	EBCDIC	Single-level data set name.
141	8D	SMF81OP4	1	binary	Options indicator 4 **Bit** **Meaning When Set** **0** TAPEDSN is in effect **1** PROTECT-ALL is in effect **2** PROTECT-ALL warning is in effect **3** ERASE-ON-SCRATCH is in effect **4** ERASE-ON-SCRATCH by SECLEVEL is in effect **5** ERASE-ON-SCRATCH for all data sets is in effect **6** Enhanced generic naming is in effect **7** Record contains a version, release, and modification number (see SMF81VRM).

Offsets		Name	Length	Format	Description
142	8E	SMF81OP5	1	binary	Options indicator 5 **Bit** **Meaning When Set** **0** Access control by program is in effect **1-3** Reserved. **4** SETROPTS NOADDCREATOR is active **5-7** Reserved.
143	8F	SMF81RPD	2	binary	System retention period in effect.
145	91	SMF81SLV	1	binary	Security level for ERASE-ON-SCRATCH in effect.
146	92	SMF81SLC	1	binary	Security level for auditing in effect.
147	93	SMF81VRM	4	EBCDIC	RACF version, release, and modification number in the form VRRM (for example, 1081 represents RACF 1.8.1 and 1100 represents RACF 1.10).
151	97	SMF81BOP	1	binary	RACF 1.9.0 SETROPTS options. **Bit** **Meaning When Set** **0** SECLABELCONTROL is in effect **1** CATDSNS is in effect **2** MLQUIET is in effect **3** MLSTABLE is in effect **4** MLS is in effect **5** MLACTIVE is in effect interval **6** GENERICOWNER is in effect **7** SECLABELAUDIT is in effect.
152	98	SMF81SIN	2	binary	Partner LU-verification session key.
154	9A	SMF81JSY	8	EBCDIC	JES NJE NAME user ID.
162	A2	SMF81JUN	8	EBCDIC	JES UNDEFINEDUSER user ID.
170	AA	SMF81BOX	1	binary	RACF 1.9.0 SETROPTS options extension **Bit** **Meaning When Set** **0** COMPATMODE is in effect **1** CATDSNS failures are in effect **2** MLS failures are in effect **3** MLACTIVE failures are in effect **4-7** Reserved.

Offsets		Name	Length	Format	Description
171	AB	SMF81PRI	3	EBCDIC	Default primary language for an installation.
174	AC	SMF81SEC	3	EBCDIC	Default secondary language for an installation.
177	B1		2		Reserved.
179	B3	SMF81OP6	1	Binary	Options indicator 6 **Bit** **Meaning When Set** **0** Mixed case password set **1** New password phrase installation exit active **2** Reserved for IBM use **3** Special characters allowed in passwords **4-7** Reserved for IBM use
180	B4		1		Reserved.
181	B5	SMF81ALG	1	Binary	Password encryption algorithm in effect **Bit** **Meaning** **0** Indicates LEGACY **1** Indicates KDFAES
182	B6	SMF81VXC	8	EBCDIC	VMXEVENT control profile in effect
190	BE	SMF81VXA	8	EBCDIC	VMXEVENT audit profile in effect
198	C6		57		Reserved.

Relocate Section:

Offsets		Name	Length	Format	Description
0	0	SMF81DTP	1	binary	Data type.
1	1	SMF81DLN	1	binary	Length of data that follows.
2	2	SMF81DTA	1-255	mixed	Data.

Record type 83: Security Events

Record type 83 is a processing record for auditing security related events. A security event can be an authentication or authorization attempt. The service detecting the event may be RACF or another z/OS component. The specific component is identified by the product section of the SMF type 83 record.

Note:

1. Subtypes 1, 2, 5 and 6 are not applicable to z/VM.
2. Subtype 3 - Used by the LDAP server to audit security-related LDAP events.
3. Subtype 4 - Used by the remote audit function provided by the LDAP server.

See *z/VM: TCP/IP Planning and Customization* for details on LDAP server audit support, and *z/VM: TCP/IP Programmer's Reference* for details on the remote audit function.

The format is:

Offsets					
Dec.	**Hex.**	**Name**	**Length**	**Format**	**Description**
0	0	SMF83LEN	2	Binary	Record length.
2	2	SMF83SEG	2	Binary	Segment descriptor.
4	4	SMF83FLG	1	Binary	System indicator **Bit** **Meaning when set** **0** Subsystem identification follows system identification **1** Subtypes used **2-7** Reserved for IBM's use.
5	5	SMF83RTY	1	Binary	Record type: 83 (X'53').
6	6	SMF83TME	4	Binary	Time of day, in hundredths of a second, that the record was moved to the SMF buffer.
10	A	SMF83DTE	4	EBCDIC	Date that the record was moved to the SMF buffer, in the form 0*cyydddF* (where *F* is the sign).
14	E	SMF83SID	4	EBCDIC	System identification (from the SID parameter).
18	12	SMF83SSI	4	EBCDIC	Subsystem identification — RACF.
22	16	SMF83TYP	2	Binary	Record subtype
24	18	SMF83TRP	2	Binary	Number of triplets.
26	1A	SMF83XXX	2		Reserved for IBM's use.
28	1C	SMF83OPD	4	Binary	Offset to product section.
32	20	SMF83LPD	2	Binary	Length of product section.
34	22	SMF83NPD	2	Binary	Number of product sections.
36	24	SMF83OD1	4	Binary	Offset to security section.
40	28	SMF83LD1	2	Binary	Length of security section.
42	2A	SMF83ND1	2	Binary	Number of security sections.
44	2C	SMF83OD2	4	Binary	Offset to relocate section.
48	30	SMF83LD2	2	Binary	Length of relocate section.
50	32	SMF83ND2	2	Binary	Number of relocate sections.
		Product section: See description below for details.			
		Security section: See description below for details.			
		Relocate sections: See description below for details.			

Product Section

The product section exists in all SMF type 83 records.

The product section in the record can be located by adding the SMF83OPD field to the beginning of the SMF record.

The product section is mapped in the following table.

Table 8. RACF SMF Type 83 record product section

Offsets					
Dec.	**Hex.**	**Name**	**Length**	**Format**	**Description**
0	0	SMF83RVN	4	EBCDIC	Product version, release, and modification level number.

Table 8. RACF SMF Type 83 record product section (continued)

Offsets					
Dec.	**Hex.**	**Name**	**Length**	**Format**	**Description**
4	4	SMF83PNM	4	EBCDIC	Product name

Security Section

The security section is common to all Record type 83 subtypes. It identifies the specific event and the result.

The information in the security section and the relocate sections provide additional information about the event.

- The user identity or identities used by the product or component for purposes of the authentication or authorization request
- The authority required for the request to succeed
- The authority the user has
- The reasons for logging the event

  1. includes the user identity used to determine why to log
  2. includes the resource used to determine why to log

Any authentication or authorization request may succeed or fail because of one of several authority checks that grant access to the system or resource. The information in the audit record is limited to the specific authority check that succeeded or failed. The audit record does not contain all of the authorities the user has or all of the authorities that could allow access to the system or resource.

The security section in the record can be located by adding the SMF83OD1 field to the beginning of the SMF record

Subtype 3 and above

Offsets					
Dec.	**Hex.**	**Name**	**Length**	**Format**	**Description**
Security Section:					
0	0	SMF83LNK	4	Binary	Value used to link several SMF 83 records to a single event.
4	4	SMF83DES	2	Binary	Descriptor flags **Bit** **Meaning when set** **0** The event is a violation **1** User is not defined to RACF **2** Reserved **3** The event is a warning **4** Record contains a version, release, and modification level number (see SMF83VRM) **5** The caller of the R_auditx service indicated always log **6-15** Reserved
6	6	SMF83EVT	1	Binary	Event code.

Offsets

Dec.	Hex.	Name	Length	Format	Description
7	7	SMF83EVQ	1	Binary	Event code qualifier.
8	8	SMF83USR	8	EBCDIC	Identifier of the user associated with this event (jobname is used if the user is not defined to RACF).
16	10	SMF83GRP	8	EBCDIC	Group to which the user was connected (stepname is used if the user is not defined to RACF).
24	18	SMF83REL	2	Binary	Reserved
26	1A	SMF83CNT	2	Binary	Reserved
28	1C	SMF83ATH	1	Binary	Authorities used for processing commands or accessing resources **Bit** **Meaning when set** **0-7** Reserved
29	1D	SMF83REA	1	Binary	Reason for logging. These flags indicate the reason RACF produced the SMF record **Bit** **Meaning when set** **0** SETROPTS AUDIT(class) — changes to this class of profile are being audited. **1** User being audited **2** SPECIAL users being audited **3** Access to the resource is being audited because of the AUDIT option (specified when profile created or altered by a RACF command), a logging request from the RACROUTE REQUEST=AUTH exit routine, or because the operator granted access during failsoft processing. **4** RACROUTE REQUEST=VERIFY or initACEE failure. **5** This command is always audited **6** Violation detected in command and CMDVIOL is in effect **7** Access to entity being audited because of GLOBALAUDIT option.
30	1E	SMF83TLV	1	Binary	Terminal level number of foreground user (zero if not available).
31	1F	SMF83ERR	1	Binary	Command processing error flag **Bit** **Meaning when set** **0** Command had error and RACF could not back out some changes **1** No profile updates were made because of error in RACF processing **2-7** Reserved
32	20	SMF83TRM	8	EBCDIC	Terminal ID of foreground user (zero if not available).
40	28	SMF83JBN	8	EBCDIC	Job name. For RACROUTE REQUEST=VERIFY and RACROUTE REQUEST=VERIFYX records for batch jobs, this field can be zero.

Offsets					
Dec.	**Hex.**	**Name**	**Length**	**Format**	**Description**
48	30	SMF83RST	4	Binary	Time, in hundredths of a second that the reader recognized the JOB statement for this job for RACROUTE REQUEST=VERIFY and RACROUTE REQUEST=VERIFYX records for batch jobs, this field can be zero.
52	34	SMF83RSD	4	Packed	Date the reader recognized the JOB statement for this job in the form 0*cyydddF* (where *F* is the sign) for RACROUTE REQUEST=VERIFY and RACROUTE REQUEST=VERIFYX records for batch jobs, this field can be zero.
56	38	SMF83UID	8	EBCDIC	User identification field from the SMF common exit parameter area. For RACROUTE REQUEST=VERIFY and RACROUTE REQUEST=VERIFYX records for batch jobs, this field can be zero.
64	40	SMF83VER	1	Binary	Reserved. Use SMF83VRM as the version indicator instead.
65	41	SMF83RE2	1	Binary	Additional reasons for logging **Bit** **Meaning when set** **0** Reserved **1** Auditing by LOGOPTIONS **2-7** Reserved
66	42	SMF83VRM	4	EBCDIC	FMID for RACF
70	46	SMF83SEC	8	EBCDIC	Security Label of the User.
78	4E	SMF83AU2	1	Binary	Reserved
79	4F	SMF83RSV	1	Binary	Reserved
80	50	SMF83US2	8	EBCDIC	Identifier of the user associated with this event.
88	58	SMF83GR2	8	EBCDIC	Group to which the user was connected.

Relocate Sections

Two types of relocate sections may be used by type 83 records-standard relocates or extended relocates. They are described below.

The start of the relocate sections in the record can be located by adding the SMF83OD2 field to the beginning of the SMF record.

The relocate sections for subtypes 3 and above use the extended relocate section format. The data types (i.e. relocate types) for the subtypes are documented with the product or component that reported the security event. Data type values of 100 and above are reserved for product or component use.

Table 9. RACF SMF record standard relocate section format

Offsets					
Dec.	**Hex.**	**Name**	**Length**	**Format**	**Description**
0	0	SMF83DTP	1	Binary	Data type
1	1	SMF83DLN	1	Binary	Length of data that follows.
2	2	SMF83DTA	1-255 (1-FF)	mixed	Data

Table 10. RACF SMF record extended relocate section format:

Offsets					
Dec.	**Hex.**	**Name**	**Length**	**Format**	**Description**
0	0	SMF83TP2	2	Binary	Data type
2	2	SMF83DL2	2	Binary	Length of data that follows.
4	4	SMF83DA2	variable	EBCDIC	Data

The relocate data type values 1-99 that appear in a SMF type 83 subtype 3 or above record are reserved for use by the RACF auditing services. The following tables lists those relocate data types that have been assigned. These data types are used only for SMF Type 83 subtype 3 records and above.

Table 11. RACF SMF Type 83 Subtype 2 and above relocates

Data Type (SMF83TP2)		Max Data Length (SMF83DL2)		Format	Audited by Event Code	Description
Dec.	**Hex.**	**Dec.**	**Hex.**			
1	1	255	FF	EBCDIC	all subtype 2 and above	Subject's distinguished name from the current ACEE
2	2	255	FF	EBCDIC	all subtype 2 and above	Issuers distinguished name from current ACEE
3	3	246	F6	EBCDIC	all subtype 2 and above	Resource name
4	4	8	8	EBCDIC	all subtype 2 and above	Class name
5	5	246	F6	EBCDIC	all subtype 2 and above	Profile name
6	6	7	7	EBCDIC	all subtype 2 and above	FMID of the product requesting event logging
7	7	255	FF	EBCDIC	all subtype 2 and above	Name of the product requesting event logging
8	8	255	FF	EBCDIC	all subtype 2 and above	Log string
9	9	8	8	Binary	all subtype 2 and above	Link value
10	A	510	1FE	EBCDIC	All subtype 2 and above	Authenticated user name
11	B	255	FF	EBCDIC	All subtype 2 and above	Authenticated user registry name
12	C	128	80	EBCDIC	All subtype 2 and above	Authenticated user host name
13	D	16	10	EBCDIC	All subtype 2 and above	Authenticated user authentication mechanism object identifier (OID)

Reformatted RACF SMF Records

For sorting purposes, the RACF report writer reformats SMF records (types 80, 81 and 83) and uses these reformatted records as input to the modules that produce the RACF reports. If you want to use the RACF report writer exit routine (ICHRSMFE) to produce additional reports or to add additional record selection criteria, you should familiarize yourself with the layouts of these reformatted records.

There are two record types—reformatted process records and reformatted status records.

Note: The layouts of reformatted process and status records are the same up to the record dependent sections.

Reformatted Process Records

RACF SMF records of type 80 become reformatted process records. These records are variable in length. Note that a RACF SMF record type 80 generated by a SETROPTS or an RVARY command also causes the creation of a reformatted status record.

The layout of the common section of the reformatted process record is:

Table 12. Format of common section of report writer reformatted records.

Offsets	Name	Length	Format	Description
0 0	RCDLEN	2	binary	Total record length
2 2	-	2	binary	Reserved
4 4	RCDRELNO	1	binary	Release of RACF
5 5	RCDREFMT	1	binary	Reformat indicator (if this byte is X'00', the record has been reformatted to the RACF Version 1 Release 6/7 format)
6 6	RCDSYSID	4	EBCDIC	System identification
10 A	RCDTYPE	1	EBCDIC	Record type (80 decimal)
11 B	RCDTIME	4	packed	Unsigned packed decimal in the form FFMMSSTH
15 F		1	EBCDIC	Reserved
16 10	RCDDATE	3	packed	Date in form YYDDDF, where F is the sign
19 13	RCDFIXLN	2	binary	Offset from the start of the record to the first relocate section
21 15	RCDCOMLN	2	binary	Offset from the start of the record to the record dependent fields
23 17	RCDCNT	2	binary	Number of relocate segments
25 19	RCDEVENT	1	binary	Event code
26 1A	RCDQUAL	1	binary	Event code qualifier
27 1B	RCD80FLG	1	binary	Descriptor flags: **Bit** **Meaning When Set** **0** This record is for security violations. **1** This record is for a job/step, not a user/group. **2** This record is truncated. **3** This record is for a warning. **4-7** Reserved.
28 1C		1	binary	Reserved
29 1D	RCDUSER	8	EBCDIC	Identifier of the user for which this event is recorded (or jobname if the user is not defined to RACF)
37 25	RCDGROUP	8	EBCDIC	Group to which the user was connected (or stepname if the user is not defined to RACF)
45 2D	RCDLOGCL	1	binary	Type of event: 1—LOGON/JOB 2—Entity access 3—RACF command

Table 12. Format of common section of report writer reformatted records. (continued)

Offsets	Name	Length	Format	Description
46 2E	RCDCLASS	8	EBCDIC	Resource class name (see Note 1). This field contains binary zeros for records written by the RVARY and SETROPTS commands.
54 36	RCDNAME	44	EBCDIC	Resource name (see Notes 1 and 6). This field contains the user ID for a LOGON/JOB; the resource name for a resource access.
98 62	RCDJOBID	8	EBCDIC	Job name
106 6A		1	EBCDIC	Reserved
107 6B	RCDDATID	3	packed	Date that the reader recognized the JOB card for this job in the form YYDDDF
110 6E	RCDTIMID	4	EBCDIC	Time that the reader recognized the JOB card for this job in the form HHMMSSTH
114 72	RCDUSRDA	8	EBCDIC	User identification field
122 7A	RCD80TRM	8	EBCDIC	Terminal identification field
130 82	RCD80TML	1	binary	Terminal level number
131 83	RCDOWNER	8	EBCDIC	Owner of the resource
139 8B	RCDUSRSM	20	EBCDIC	User name
159 9F	RCDVRM	4	EBCDIC	Release, version and modification number
163 A3	RCDSEC	8	EBCDIC	User's SECLABEL
171 AB	RCDLINK	4	binary	LINK to connect data sets affected by a SECLABEL change with RACF command (ALTDSD, ADDSD, DELDSD) that caused the change.
175 AF	RCDSTYPE	2	binary	SMF record subtype
177 B1	RCDNAMEO	2	binary	See Note 6. Offset in variable section to relocate section type if entity name is greater than 44 characters or X'7FFF' if resource name is less than or equal to 44 characters.

For process records, the record-dependent section is:

Offsets	Name	Length	Format	Description
0 0	RCD80ATH	1	binary	Authority used: **Bit** **Meaning When Set** **0** Normal authority **1** SPECIAL attribute **2** OPERATIONS attribute **3** AUDITOR attribute **4** Exit routine granted authority **5** Failsoft processing **6** Bypassed-user ID=*BYPASS* **7** Trusted attribute

Offsets	Name	Length	Format	Description
1 1	RCD80REA	2	binary	Reason for logging: **Bit** **Meaning When Set** **0** Class being audited **1** User being audited **2** Special user being audited **3** Resource being audited, installation-requested logging in effect, or failsoft processing **4** RACINIT failures being audited **5** Command always causes auditing **6** Command violations being audited **7** Audited because GLOBALAUDIT option in effect **8** SECLEVEL audit **9-15** Reserved
3 3	RCD80ERR	1	binary	Error indicators: **Bit** **Meaning When Set** **0** Command could not recover **1** Profile not altered **2-7** Reserved
4 4	RCDQUAL1	8	EBCDIC	Qualifier for old data set name (see Note 2)
12 C	RCDQUAL2	8	EBCDIC	Qualifier for new data set name (see Note 3)
20 14	RCDDLEV	1	binary	Data set level number (see Note 4)
21 15	RCDDINT	1	binary	Access authority requested: (see Note 4) **Bit** **Access Authority** **0** ALTER **1** CONTROL **2** UPDATE **3** READ **4-7** Reserved.

Offsets	Name	Length	Format	Description
22 16	RCDDALWD	1	binary	Access authority allowed: (see Note 4) **Bit** **Access Authority** **0** ALTER **1** CONTROL **2** UPDATE **3** READ **4** NONE **5** EXECUTE **6-7** Reserved
23 17	RCDDVOL	6	EBCDIC	Volume serial (see Note 4)
29 1D	RCDDOLDV	6	EBCDIC	OLDVOL volume serial (see Note 4)
35 23	RCD80GNS	1	binary	1=Generic name specified
36 24	RCD80GSP	1	binary	1=Generic name specified on FROM keyword of PERMIT
37 25	RCD80RRF	1	binary	1=The **old** name of the RACDEF-renamed data set from data type 33 relocate section
38 26	RCD80RRT	1	binary	1=The **new** name of the RACDEF-renamed data set from data type 33 relocate section
39 27	RCDGENAM	44	EBCDIC	Generic profile used or generic resource name (see Note 7)
83 53	RCDGNNMF	44	EBCDIC	Generic profile used on RACDEF RENAME or generic resource name on RACDEF RENAME Relocate Section: (See Notes 5 and 8)
127 7F	RCDGENAO	2	binary	See Note 7
129 81	RCDGNNMO	2	binary	See Note 8
Variable Relocate Section Map				
+0 0	RCDDTYPE	1	binary	Data type
+1 1	RCDDLGT	1	binary	Length of data that follows
+2 2	RCDDATA	variable	mixed	Data

Note 1: In order to support sorting by resource class name and resource name for the list report, the RACF report writer ensures that these fields contain valid names. The following table indicates the resource class names and the resource names assigned by the RACF report writer for each of the event codes in RCDEVENT. (Uppercase letters indicate that the value appears as shown, lowercase letters identify the field in the SMF type 80 record from which the name is obtained, and a number in parentheses identifies the relocate section in the SMF type 80 record from which the name is obtained.)

If RCDEVENT is	Resource Class Name	Resource Name
1	USER	user ID (SMF80USR)
2	class name (17)	resource name (1)
3	class name (17)	resource name (1)
4	class name (17)	resource name (1)

If RCDEVENT is	Resource Class Name	Resource Name
5	class name (17)	resource name (1)
6	class name (17)	resource name (1)
7	class name (17)	resource name (1)
8	DATASET	data set name (6)
9	GROUP	group name (6)
10	USER	user ID (6)
11	DATASET	data set name (6)
12	GROUP	group name (6)
13	USER	user ID (6)
14	USER	user ID (6)
15	DATASET	data set name (6)
16	GROUP	group name (6)
17	USER	user ID (6)
18	USER	user ID (6)
19	class name (17)	resource name (9)
20	class name (17)	resource name (9)
21	class name (17)	resource name (9)
22	class name (17)	resource name (9)
23	USER	user ID (6)
24	none	none
25	none	none

Note 2: The RACF report writer compares this field to the DSQUAL keyword specified on the EVENT subcommand. The report writer initializes RCDQUAL1 to the high-level qualifier of the old data set name found in RCDNAME at offset 41 (29 hex) of this record. The RACF report writer exit routine, ICHRSMFE, can modify this field.

Note 3: The RACF report writer compares this field to the NEWDSQUAL keyword specified on the EVENT subcommand. The report writer initializes RCDQUAL to the high-level qualifier of the new data set name found in the relocate section for data type 2 (SMF80DTP = 2). The RACF report writer exit routine, ICHRSMFE, can modify this field.

Note 4: This field is present for event codes 2-7 (SMF80EVT=2 through SMF80EVT=7) only.

Note 5: See “Table of Event Codes and Event Code Qualifiers” on page 36 and “Table of Relocate Section Variable Data” on page 47 earlier in this chapter for a further explanation of these event codes and data types.

Note 6: With RACF 1.9 or later, entity names can be a maximum of 254 characters. Entity names containing 45–254 characters are referred to as *long* names. Field RCDNAME cannot be expanded in order to support existing reformatted records. Long resource names are handled as follows:

Field RCDNAMEO will contain the offset in the variable section of the reformatted record of relocate type which contains the long resource name.

Field RCDNAMEO will be hex X'7FFF' if the resource name is less than or equal to 44 characters in length.

Note 7: With RACF 1.9 or later, entity names can be a maximum of 254 characters. Field RCDGENAM cannot be expanded in order to support existing reformatted records. Long resource names are handled as follows:

Field RCDGENAO will contain the offset in the variable section of the reformatted record of relocate type which contains the long resource name.

Field RCDGENAO will be hex X'7FFF' if the resource name is less than or equal to 44 characters in length.

Note 8: With RACF 1.9 or later, entity names can be a maximum of 254 characters. Field RCDGNNMF cannot be expanded in order to support existing reformatted records. Long resource names are handled as follows:

Field RCDGNNMO will contain the offset in the variable section of the reformatted record of relocate type which contains the long resource name.
Field RCDGNNMO will be hex X'7FFF' if the resource name is less than or equal to 44 characters in length.

Reformatted Status Records

RACF SMF record types 80 (only those generated by the SETROPTS or RVARY command) and 81 become reformatted status records.

The layout of the common section of the reformatted status record is on pageTable 12 on page 136.

For status records, the record-dependent section is:

Table 13. Record-dependent section of status records.

Offsets	Name	Length	Format	Description
0 00	RCDRACFD	44	EBCDIC	Name of the RACF database for this IPL
44 2C	RCDRACFV	6	EBCDIC	Volume identification of RACF database
50 32	RCDRACFU	3	EBCDIC	Unit name of RACF database
53 35	RCD81FLG	1	binary	Options indicators: **Bit** **Meaning When Set** **0** No RACINIT statistics are recorded **1** No DATASET statistics are recorded **2** RACINIT preprocessing exit routine, ICHRIX01, is active **3** RACHECK preprocessing exit routine, ICHRCX01, is active **4** RACDEF preprocessing exit routine, ICHRDX01, is active **5** RACINIT postprocessing exit routine, ICHRIN02, is active **6** RACHECK postprocessing exit routine, ICHRCX02, is active **7** New password exit routine, ICHPWX01, is active
54 36	RCDUVOL	6	EBCDIC	Volume identification of UADS data set
60 3C	RCDUDSN	44	EBCDIC	Data set name of the UADS data set for this IPL

Table 13. Record-dependent section of status records. (continued)

Offsets	Name	Length	Format	Description
104 68	RCD81FG2	1	binary	Options indicators: **Bit** **Meaning When Set** **0** No tape volume statistics are recorded **1** No DASD volume statistics are recorded **2** No terminal statistics are recorded **3** Command exit routine ICHCNX00 is active **4** Command exit routine ICHCCX00 is active **5** ADSP is not active **6** Encryption exit routine, ICHDEX01, is active **7** Naming convention table, ICHNCV00, is present
105 69	RCD81OP3	1	binary	Options indicators: **Bit** **Meaning When Set** **0** Tape volume protection in effect **1** No duplicate data set protection in effect **2** DASD volume protection in effect **3** Reserved **4** FRACHECK exit routine is active **5** RACLIST pre/postprocessing exit routine is active **6** RACLIST selection exit routine is active **7** RACDEF postprocessing exit routine is active

Table 13. Record-dependent section of status records. (continued)

Offsets	Name	Length	Format	Description
106 6A	RCD81AOP	1	binary	Options indicators: **Bit** **Meaning When Set** **0** Log all users **1** Log all groups **2** Log data set class **3** Log tape volume class **4** Log DASD volume class **5** Log terminal class **6** Log command violations **7** Log special users
107 6B	RCD81TMO	1	binary	Options indicators: **Bit** **Meaning When Set** **0** Terminal authorization checking in effect **1** UACC for undefined terminals is NONE **2** REALDSN is in effect **3** JES-XBMALLRACF is in effect **4** JES-EARLYVERIFY is in effect **5** JES-BATCHALLRACF is in effect **6** FRACHECK postprocessing exit is active **7** Reserved
108 6C	RCD81PIV	1	binary	Maximum password interval
109 6D	RCD81MFG	1	binary	Model flags: **Bit** **Meaning When Set** **0** Model—GDG **1** Model—USER **2** Model—GROUP **3-7** Reserved

Table 13. Record-dependent section of status records. (continued)

Offsets	Name	Length	Format	Description
110 6E	RCD81MSF	1	binary	Miscellaneous processing flags: **Bit** **Meaning When Set** **0** GRPLIST active **1** Generic profile checking in effect for data set **2** GENCMD in effect for data set class **3** ADSP attribute bypassed **4-7** Reserved
111 6F	RCD81IFG	1	binary	Internal processing flags: **Bit** **Meaning When Set** **0** The SETROPTS command caused RACFRW to generate this record **1** The RVARY command caused RACFRW to generate this record **2** RACF was varied active by RVARY command **3** This record is incomplete (truncated) **4** RVARY SWITCH was issued **5-7** Reserved
112 70	RCD81QL	8	char	Single level data set name prefix
120 78	RCD81A02	1	binary	Options indicator **Bit** **Meaning When Set** **0** Log OPERATIONS user **1-7** Reserved **2** RACF was varied active by RVARY command.

Table 13. Record-dependent section of status records. (continued)

Offsets	Name	Length	Format	Description
121 79	RCD81OP4	1	binary	Options indicators **Bit** **Meaning When Set** **0** Tape DSN active **1** PROTECTALL active **2** PROTECTALL warning **3** Erase-on-scratch **4** Erase by SECLEVEL **5** Erase all files **6-7** Reserved
122 7A	RCD81OP5	1	binary	Options indicators **Bit** **Meaning When Set** **0** Program control active **1-7** Reserved
124 7C	RCD81RPD	2	binary	Data set retention period
126 7E	RCD81SLV	1	char	SECLEVEL number
127 7F	RCD81SLC	1	binary	SECLEVEL for auditing number
128 80	RCD81BOP	1	binary	B1 security options **Bit** **Meaning When Set** **0** SECLABELCONTROL active **1** CATDSNS active **2** MLQUIET active **3** MLSTABLE active **4** MLS active **5** MLACTIVE active **6** GENERICOWNER active **7** SECLABELAUDIT active
129 81	RCD81SIN	2	binary	SESSION INTERVAL
131 83	RCD81SYS	8	char	User ID for JES SYSOUTNAME
139 8B	RCD81UND	8	char	User ID for JES undefined user

Table 13. Record-dependent section of status records. (continued)

Offsets	Name	Length	Format	Description
147 93	RCD81BOX	1	binary	B1 security options extension byte **Bit** **Meaning When Set** **0** COMPATMODE **1** CATDSNS failures **2** MLS failures **3** MLACTIVE failures **4-7** Reserved
148 94	RCD81PRI	3	EBCDIC	Primary language default
151 97	RCD81SEC	3	EBCDIC	Secondary language default

Offsets	Name	Length	Format	Description
+0(0)	RCDDTYPE	1	binary	Data type
+1(1)	RCDDLGT	1	binary	Length of data that follows
+2(2)	RCDDATA	variable	mixed	Data

Note: Only data types (SMF80DTP) 21, 30, 32, 34, 35 or 36 are generated for a reformatted status record. See "Table of Relocate Section Variable Data" on page 47 earlier in this chapter for a further explanation of these data types.

Chapter 4. RACF SMF Data Unload Record Formats

Notes to Reader
The SMF data unload utility can unload SMF data in two formats: • A tabular format, suitable for export to a relational database manager. This chapter documents that format. • An eXtended Markup Language (XML) document, which can be rendered into different formats such as Web pages (HTML). An installation can write applications that interpret the data to generate custom reports. For information about how to convert the field names in the tabular format to XML tags, see "XML grammar" on page 261. • Not all record types (nor individual fields within record types) documented herein are created on z/VM. They are documented to maintain compatibility and consistency with the z/OS version of RACF.

Record Format

The following sections contain a detailed description of the records that are produced by the RACF SMF data unload utility (RACFADU). For information on running RACFADU, see *z/VM: RACF Security Server Auditor's Guide*. The RACF SMF data unload format records represent the security relevant SMF data. These records are in a format suitable for export to the relational database manager of an installation's choice. For more information on using these records with a database manager, see *z/VM: RACF Security Server Auditor's Guide*.

Each row in the tabular description of the records that are produced contains five pieces of information:

1. Descriptive name for the field (Field Name)
2. Type of field (Type)

   Char
   Character data

   Integer
   EBCDIC numeric data

   Time
   A time value, in the form *hh:mm:ss*

   Date
   A date value, in the form *yyyy-mm-dd*

   Yes/No
   Flag data, having the value YES or N0
3. Starting position for the field (Start)
4. Ending position for the field (End)
5. Free-form description of the field, which may contain the valid value constraints (Comments).

The Format of the Header Portion of the Unloaded SMF Type 80 Data

Each RACF SMF data unload record that is produced consists of two parts:

1. A header section, which contains common information such as the date and time stamp, user ID, and system identification
2. An event-specific information section

Table 14 on page 148 describes the format of the header portion of the record. The sections that follow describe the event-specific information. The header portion of each record reflects the information in the base SMF record. The extensions reflect the relocate section data for a specific event code. All relocate sections are *not* created for every event code, so fields in the event-specific information may contain blanks.

The string *col-id* is replaced by the column identifier for each record created. See Table 15 on page 150 for a list of valid column identifiers.

Table 14. Format of the Header Portion of the Unloaded SMF Records

Field Name	Type	Length	Position		Comments
			Start	End	
*col-id*_EVENT_TYPE	Char	8	1	8	Type of event that is described. Valid values are shown in Table 15 on page 150. A numeric value indicates that the event code was not translated. Only header information is created for records that have an untranslated event code.
*col-id*_EVENT_QUAL	Char	8	10	17	A qualification of the type of event that is being described. Valid values are shown in the tables that accompany each of the record extension descriptions.
*col-id*_TIME_WRITTEN	Time	8	19	26	Time that the record was written to SMF.
*col-id*_DATE_WRITTEN	Date	10	28	37	Date that the record was written to SMF.
*col-id*_SYSTEM_SMFID	Char	4	39	42	SMF system ID of the system from which the record originates.
*col-id*_VIOLATION	Yes/No	4	44	47	Does this record represent a violation?
*col-id*_USER_NDFND	Yes/No	4	49	52	Was this user not defined to RACF?
*col-id*_USER_WARNING	Yes/No	4	54	57	Was this record created because of WARNING?
*col-id*_EVT_USER_ID	Char	8	59	66	User ID associated with the event.
*col-id*_EVT_GRP_ID	Char	8	68	75	Group ID associated with the event. See note after table.
*col-id*_AUTH_NORMAL	Yes/No	4	77	80	Was normal authority checking a reason for access being allowed?
*col-id*_AUTH_SPECIAL	Yes/No	4	82	85	Was special authority checking a reason for access being allowed?
*col-id*_AUTH_OPER	Yes/No	4	87	90	Was operations authority checking a reason for access being allowed?
*col-id*_AUTH_AUDIT	Yes/No	4	92	95	Was auditor authority checking a reason for access being allowed?
*col-id*_AUTH_EXIT	Yes/No	4	97	100	Was exit checking a reason for access being allowed?
*col-id*_AUTH_FAILSFT	Yes/No	4	102	105	Was failsoft checking a reason for access being allowed?
*col-id*_AUTH_BYPASS	Yes/No	4	107	110	Was the use of the user ID *BYPASS* a reason for access being allowed?
*col-id*_AUTH_TRUSTED	Yes/No	4	112	115	Was trusted authority checking a reason for access being allowed?
*col-id*_LOG_CLASS	Yes/No	4	117	120	Was SETR AUDIT(class) checking a reason for this event to be recorded?
*col-id*_LOG_USER	Yes/No	4	122	125	Was auditing requested for this user?
*col-id*_LOG_SPECIAL	Yes/No	4	127	130	Was auditing requested for access granted due to the SPECIAL privilege?
*col-id*_LOG_ACCESS	Yes/No	4	132	135	Did the profile indicate audit, did FAILSOFT processing allow access, or did the RACROUTE REQUEST=AUTH exit indicate auditing?

Table 14. Format of the Header Portion of the Unloaded SMF Records (continued)

Field Name	Type	Length	Position		Comments
			Start	End	
*col-id*_LOG_RACINIT	Yes/No	4	137	140	Did the RACINIT fail?
*col-id*_LOG_ALWAYS	Yes/No	4	142	145	Is this command always audited?
*col-id*_LOG_CMDVIOL	Yes/No	4	147	150	Was this event audited due to CMDVIOL?
*col-id*_LOG_GLOBAL	Yes/No	4	152	155	Was this event audited due to GLOBALAUDIT?
*col-id*_TERM_LEVEL	Integer	3	157	159	The terminal level associated with this audit record.
*col-id*_BACKOUT_FAIL	Yes/No	4	161	164	Did RACF fail in backing out the data?
*col-id*_PROF_SAME	Yes/No	4	166	169	Did a RACF error cause the profile not to be changed?
*col-id*_TERM	Char	8	171	178	The terminal associated with the event. See note after table.
*col-id*_JOB_NAME	Char	8	180	187	The job name associated with the event. This field is not relevant when the audit record is created on a z/VM system and will contain blanks.
*col-id*_READ_TIME	Time	8	189	196	The time that the job entered the system. This field is not relevant when the audit record is created on a z/VM system and will contain blanks.
*col-id*_READ_DATE	Date	10	198	207	The date that the job entered the system. This field is not relevant when the audit record is created on a z/VM system and will contain blanks.
*col-id*_SMF_USER_ID	Char	8	209	216	User ID from SMF common area. This value is managed by SMF and the SMF processing exits. This field is not relevant when the audit record is created on a z/VM system and will contain blanks.
*col-id*_LOG_LEVEL	Yes/No	4	218	221	Was this event audited due to SECLEVEL auditing?
*col-id*_LOG_VMEVENT	Yes/No	4	223	226	Was this event audited due to VMEVENT auditing?
*col-id*_LOG_LOGOPT	Yes/No	4	228	231	Was this event audited due to SETR LOGOPTIONS auditing?
*col-id*_LOG_SECL	Yes/No	4	233	236	Was this event audited due to SETR SECLABELAUDT auditing?
*col-id*_LOG_COMPATM	Yes/No	4	238	241	Was this event audited due to SETR COMPATMODE auditing?
*col-id*_LOG_APPLAUD	Yes/No	4	243	246	Was this event audited due to SETR APPLAUDIT?
*col-id*_LOG_NONOMVS	Yes/No	4	248	251	Did this user try to use OpenExtensions without being defined as an OpenExtensions user (that is, is the user's OMVS segment in the RACF data base missing)?
*col-id*_LOG_OMVSNPRV	Yes/No	4	253	256	The service that was requested requires that the user be the OpenExtensions super-user.
*col-id*_AUTH_OMVSSU	Yes/No	4	258	261	Was the OpenExtensions super-user authority used to grant the request?
*col-id*_AUTH_OMVSSYS	Yes/No	4	263	266	Was the request granted because the requester was OpenExtensions itself?
*col-id*_USR_SECL	Char	8	268	275	The SECLABEL associated with this user.
*col-id*_RACF_VERSION	Char	4	277	280	The version of RACF on the system that audited the event.

Table 14. Format of the Header Portion of the Unloaded SMF Records (continued)					
Field Name	**Type**	**Length**	**Position**		**Comments**
			Start	**End**	
Note: Fields *col-id*_EVT_GRP_ID and *col-id*_TERM contain blanks if the audit record is created due to: 1. VMXEVENT auditing 2. An OpenExtensions z/VM access attempt, and the client has logged off when the audit request gets to the RACFVM service machine In addition, the following fields are not relevant when the audit record is created on a z/VM system, and will therefore be blank: *col-id*_JOB_NAME *col-id*_READ_DATE *col-id*_READ_TIME *col-id*_SMF_USER_ID					

Event Codes

The RACF SMF data unload format records represent the audit information for each type of auditable event. Table 15 on page 150 contains a list of all of the supported event codes.

Column Name
Description

Event Code Name
Name of the event code

Column ID
Shortened name that is used in the column name of fields that are a part of the event code record

Event Code Number
The number assigned to this event code by RACF

Description
A description of the event code.

Where Described
Where you can find the record definitions.

Table 15. Event Codes and Descriptions				
Event Code Name	**Column ID**	**Event Code Number**	**Description**	**Where Described**
JOBINIT	INIT	01	Job initiation	"The Format of the JOBINIT Record Extension" on page 152
ACCESS	ACC	02	Resource access, other than file or directory.	"The Format of the ACCESS Record Extension" on page 155
ADDVOL	ADV	03	ADDVOL/CHGVOL	"The Format of the ADDVOL Record Extension" on page 157
RENAMEDS	REN	04	Rename dataset, SFS file, or SFS directory	"The Format of the RENAMEDS Record Extension" on page 159
DELRES	DELR	05	Delete resource	"The Format of the DELRES Record Extension" on page 160
DELVOL	DELV	06	Delete volume	"The Format of the DELVOL Record Extension" on page 162
DEFINE	DEF	07	Define resource	"The Format of the DEFINE Record Extension" on page 163
ADDSD	AD	08	ADDSD command	"The Format of the ADDSD Record Extension" on page 164
ADDGROUP	AG	09	ADDGROUP command	"The Format of the ADDGROUP Record Extension" on page 166
ADDUSER	AU	10	ADDUSER command	"The Format of the ADDUSER Record Extension" on page 167
ALTDSD	ALD	11	ALTDSD command	"The Format of the ALTDSD Record Extension" on page 168
ALTGROUP	ALG	12	ALTGROUP command	"The Format of the ALTGROUP Record Extension" on page 170
ALTUSER	ALU	13	ALTUSER command	"The Format of the ALTUSER Record Extension" on page 171
CONNECT	CON	14	CONNECT command	"The Format of the CONNECT Record Extension" on page 172
DELDSD	DELD	15	DELDSD command	"The Format of the DELDSD Record Extension" on page 174

Table 15. Event Codes and Descriptions (continued)

Event Code Name	Column ID	Event Code Number	Description	Where Described
DELGROUP	DG	16	DELGROUP command	"The Format of the DELGROUP Record Extension" on page 175
DELUSER	DU	17	DELUSER command	"The Format of the DELUSER Record Extension" on page 176
PASSWORD	PWD	18	PASSWORD command	"The Format of the PASSWORD Record Extension" on page 177
PERMIT	PERM	19	PERMIT, PERMDIR, or PERMFILE command	"The Format of the PERMIT Record Extension" on page 179
RALTER	RALT	20	RALTER, ALTDIR, or ALTFILE command	"The Format of the RALTER Record Extension" on page 180
RDEFINE	RDEF	21	RDEFINE, ADDDIR, or ADDFILE command	"The Format of the RDEFINE Record Extension" on page 181
RDELETE	RDEL	22	RDELETE, DELDIR, or DELFILE command	"The Format of the RDELETE Record Extension" on page 183
REMOVE	REM	23	REMOVE command	"The Format of the REMOVE Record Extension" on page 184
SETROPTS	SETR	24	SETROPTS command	"The Format of the SETROPTS Record Extension" on page 185
RVARY	RVAR	25	RVARY command	"The Format of the RVARY Record Extension" on page 186
APPCLU	APPC	26	APPC session	"The Format of the APPCLU Record Extension" on page 187
GENERAL	GEN	27	General purpose	"The Format of the General Event Record Extension" on page 189
DIRSRCH	DSCH	28	Directory Search	"The Format of the Directory Search Record Extension" on page 190
DACCESS	DACC	29	Check access to a directory	"The Format of the Check Directory Access Record Extension" on page 192
FACCESS	FACC	30	Check access to file	"The Format of the Check File Access Record Extension" on page 194
CHAUDIT	CAUD	31	Change audit options	"The Format of the Change Audit Record Extension" on page 196
CHDIR	CDIR	32	Change current directory	"The Format of the Change Directory Record Extension" on page 199
CHMOD	CMOD	33	Change file mode	"The Format of the Change File Mode Record Extension" on page 200
CHOWN	COWN	34	Change file ownership	"The Format of the Change File Ownership Record Extension" on page 203
CLRSETID	CSID	35	Clear SETID bits for a file	"The Format of the Clear SETID Bits Record Extension" on page 204
EXESETID	ESID	36	EXEC with SETUID/SETGID	"The Format of the EXEC SETUID/SETGID Record Extension" on page 206
GETPSENT	GPST	37	Get OpenExtensions process entry	"The Format of the GETPSENT Record Extension" on page 208
INITOEDP	IOEP	38	Initialize OpenExtensions process	"The Format of the Initialize OpenExtensions Process Record" on page 209
TERMOEDP	TOEP	39	OpenExtensions process complete	"The Format of the OpenExtensions Process Completion Record" on page 211
KILL	KILL	40	Terminate a process	"The Format of the KILL Process Record Extension" on page 212
LINK	LINK	41	LINK	"The Format of the LINK Record Extension" on page 214
MKDIR	MDIR	42	Make directory	"The Format of the MKDIR Record Extension" on page 215
MKNOD	MNOD	43	Make node	"The Format of the MKNOD Record Extension" on page 218
MNTFSYS	MFS	44	Mount a file system	"The Format of the Mount File System Record Extension" on page 221
OPENFILE	OPEN	45	Open a new file	"The Format of the OPENFILE Record Extension" on page 223
PTRACE	PTRC	46	PTRACE authority checking	"The Format of the PTRACE Record Extension" on page 225
RENAMEF	RENF	47	Rename file	"The Format of the Rename File Record Extension" on page 227
RMDIR	RDIR	48	Remove directory	"The Format of the RMDIR Record Extension" on page 229
SETEGID	SEGI	49	Set effective GID	"The Format of the SETEGID Record Extension" on page 230
SETEUID	SEUI	50	Set effective UID	"The Format of the SETEUID Record Extension" on page 232
SETGID	SGI	51	Set GID	"The Format of the SETGID Record Extension" on page 233
SETUID	SUI	52	Set UID	"The Format of the SETUID Record Extension" on page 235

Table 15. Event Codes and Descriptions (continued)

Event Code Name	Column ID	Event Code Number	Description	Where Described
SYMLINK	SYML	53	SYMLINK	"The Format of the SYMLINK Record Extension" on page 236
UNLINK	UNL	54	UNLINK	"The Format of the UNLINK Record Extension" on page 238
UMNTFSYS	UFS	55	Unmount file system	"The Format of the Unmount File System Record Extension" on page 239
CHKFOWN	CFOW	56	Check file owner	"The Format of the Check File Owner Record Extension" on page 241
CHKPRIV	CPRV	57	Check privilege	"The Format of the Check Privilege Record Extension" on page 243
OPENSTTY	OSTY	58	Open slave TTY	"The Format of the Open Slave TTY Record Extension" on page 244
RACLINK	RACL	59	RACLINK command	"The Format of the RACLINK Command Record Extension" on page 245
IPCCHK	ICHK	60	Check IPC access	"The Format of the IPCCHK Access Record Extension" on page 247
IPCGET	IGET	61	IPCGET	"The Format of the IPCGET Access Record Extension" on page 249
IPCCTL	ICTL	62	IPCCTL	"The Format of the IPCCTL Access Record Extension" on page 251
SETGROUP	SETG	63	SETGROUP	"The Format of the SETGROUP Process Record" on page 253
CKOWN2	CKO2	64	Check owner, two files	"The Format of the Check Owner, Two Files Record Extension" on page 255

The Format of the JOBINIT Record Extension

Table 16 on page 152 describes the format of a record that is created by the RACINIT function, which occurs for user logons, batch job initiations, and at other times during the life of a unit of work.

Table 16. Format of the JOBINIT Record Extension

Field Name	Type	Length	Position		Comments
			Start	End	
INIT_APPL	Char	8	282	289	Application name specified on the RACROUTE REQUEST=AUTH or RACROUTE REQUEST=VERIFY.
INIT_LOGSTR	Char	255	291	545	LOGSTR= data from the RACROUTE
INIT_BAD_JOBNAME	Char	8	547	554	The invalid job name that was processed.
INIT_USER_NAME	Char	20	556	575	The name associated with the user ID.
INIT_UTK_ENCR	Yes/No	4	577	580	Is the UTOKEN associated with this user encrypted?
INIT_UTK_PRE19	Yes/No	4	582	585	Is this a token for a release earlier than RACF 1.9?
INIT_UTK_VERPROF	Yes/No	4	587	590	Is the VERIFYX propagation flag set?
INIT_UTK_NJEUNUSR	Yes/No	4	592	595	Is this the NJE undefined user?
INIT_UTK_LOGUSR	Yes/No	4	597	600	Is UAUDIT specified for this user?
INIT_UTK_SPECIAL	Yes/No	4	602	605	Is this a RACF SPECIAL user?
INIT_UTK_DEFAULT	Yes/No	4	607	610	Is this a default token?
INIT_UTK_UNKNUSR	Yes/No	4	612	615	Is this an undefined user?
INIT_UTK_ERROR	Yes/No	4	617	620	Is this user token in error?
INIT_UTK_TRUSTED	Yes/No	4	622	625	Is this user a part of the trusted computing base (TCB)?
INIT_UTK_SESSTYPE	Char	8	627	634	The session type of this session. See *z/VM: Security Server RACROUTE Macro Reference* for a description of the valid values for session type. A null session type results in the unloading of blanks.
INIT_UTK_SURROGAT	Yes/No	4	636	639	Is this a surrogate user?
INIT_UTK_REMOTE	Yes/No	4	641	644	Is this a remote job?

Table 16. Format of the JOBINIT Record Extension (continued)

Field Name	Type	Length	Position		Comments
			Start	End	
INIT_UTK_PRIV	Yes/No	4	646	649	Is this a privileged user ID?
INIT_UTK_SECL	Char	8	651	658	The SECLABEL of the user.
INIT_UTK_EXECNODE	Char	8	660	667	The execution node of the work.
INIT_UTK_SUSER_ID	Char	8	669	676	The submitting user ID.
INIT_UTK_SNODE	Char	8	678	685	The submitting node.
INIT_UTK_SGRP_ID	Char	8	687	694	The submitting group ID.
INIT_UTK_SPOE	Char	8	696	703	The port of entry.
INIT_UTK_SPCLASS	Char	8	705	712	Class of the POE. Valid values are "TERMINAL", "CONSOLE", "JESINPUT", and "APPCPORT".
INIT_UTK_USER_ID	Char	8	714	721	User ID associated with the record.
INIT_UTK_GRP_ID	Char	8	723	730	Group ID associated with the record.
INIT_UTK_DFT_GRP	Yes/No	4	732	735	Is a default group assigned?
INIT_UTK_DFT_SECL	Yes/No	4	737	740	Is a default SECLABEL assigned?
INIT_APPC_LINK	Char	16	742	757	A key to link together audit records for a user's APPC transaction processing work.
INIT_ACEE_VLF	Y/N	4	4540	4543	[z/OS only] The ACEE was created from the VLF cache.
INIT_MFA_USER	Y/N	4	4545	4548	The user is an MFA user.
INIT_MFA_FALLBACK	Y/N	4	4550	4553	The user is allowed to issue LOGON PWFALLBACK.
INIT_MFA_UNAVAIL	Y/N	4	4555	4558	No decision made because MFA server was unavailable.
INIT_MFA_PWD_EXPIRED	Y/N	4	4560	4563	[z/OS only] IBM MFA requested that RACROUTE REQUEST=VERIFY return the password expired return code.
INIT_MFA_NPWD_INV	Y/N	4	4565	4568	[z/OS only] IBM MFA requested that RACROUTE REQUEST=VERIFY return the password invalid return code.
INIT_MFA_PART_SUCC	Y/N	4	4570	4573	[z/OS only] IBM MFA requested that RACROUTE REQUEST=VERIFY return the password invalid return code but not increment the password revoke count (partial success).
INIT_RESERVED_01	Y/N	4	4575	4578	Reserved for IBM use.
INIT_PASSWORD_EVAL	Y/N	4	4580	4583	The supplied password was evaluated.
INIT_PASSWORD_SUCC	Y/N	4	4585	4588	The supplied password was evaluated successfully.
INIT_PHRASE_EVAL	Y/N	4	4590	4593	The supplied password phrase was evaluated.
INIT_PHRASE_SUCC	Y/N	4	4595	4598	The supplied password phrase was evaluated successfully.
INIT_PASSTICKET_EVAL	Y/N	4	4600	4603	The supplied password was evaluated as a PassTicket.
INIT_PASSTICKET_SUCC	Y/N	4	4605	4608	The supplied password was evaluated successfully as a PassTicket.
INIT_MFA_SUCC	Y/N	4	4610	4613	The supplied password/phrase was evaluated successfully as multi-factor data.
INIT_MFA_FAIL	Y/N	4	4615	4618	The supplied password/phrase was evaluated unsuccessfully as multi-factor data.
INIT_AUTH_RSN1	Char	8	4620	4627	[z/OS only] Authentication reason, Part 1. Expressed as a hexadecimal number.

Table 16. Format of the JOBINIT Record Extension (continued)					
Field Name	**Type**	**Length**	**Position**		**Comments**
			Start	**End**	
INIT_AUTH_RSN2	Char	8	4629	4636	[z/OS only] Authentication reason, Part 2. Expressed as a hexadecimal number.

Event Qualifiers for JOBINIT Records

The event qualifiers that may be associated with a JOBINIT event are shown in Table 17 on page 154.

Table 17. Event Code Qualifiers for JOBINIT Records		
Event Qualifier	**Event Number**	**Event Description**
SUCCESS	--	Successful initiation (from type 30 record)
TERM	--	Successful termination (from type 30 record)
SUCCESSI	00	Successful initiation.
INVPSWD	01	Not a valid password.
INVGRP	02	Not a valid group.
INVOID	03	Not a valid OIDCARD.
INVTERM	04	Not a valid terminal.
INVAPPL	05	Not a valid application.
REVKUSER	06	User has been revoked.
REVKAUTO	07	User automatically revoked due to excessive password and password phrase attempts.
SUCCESST	08	Successful termination.
UNDFUSER	09	User not defined to RACF.
INSSECL	10	Insufficient SECLABEL.
NASECL	11	Not authorized to SECLABEL.
RACINITI	12	Successful RACROUTE REQUEST=VERIFY,ENVIR=CREATE.
RACINITD	13	Successful RACROUTE REQUEST=VERIFY,ENVIR=DELETE.
MOREAUTH	14	More authority required.
RJENAUTH	15	RJE not authorized.
SURROGTI	16	Surrogate class inactive.
SUBNATHU	17	Submitter not authorized by user.
SUBNATHS	18	Submitter not authorized by SECLABEL.
USERNJOB	19	User not authorized to the job.
WINSSECL	20	Warning: Insufficient SECLABEL.
WSECLM	21	Warning: SECLABEL missing from job.
WNASECL	22	Warning: Not authorized to SECLABEL.
SECLNCM	23	SECLABELs not compatible.
WSECLNCM	24	Warning: SECLABELs not compatible.
PWDEXPR	25	Current password has expired.
INVNPWD	26	Not a valid new password.
EXITFAIL	27	Failed by installation exit.
GRPARVKD	28	Group access revoked.

Table 17. Event Code Qualifiers for JOBINIT Records (continued)

Event Qualifier	Event Number	Event Description
OIDREQD	29	OIDCARD required.
NJENAUTH	30	NJE job not authorized.
WUKNUPRP	31	Warning: Undefined user from trusted node propagated.
SUCCESSP	32	Successful initiation using a PassTicket.
PTKTREPL	33	PassTicket replay attempted.
REVKINAC	35	User automatically revoked due to inactivity.
INVPHRS	36	Password phrase is not valid.
INVNPHRS	37	New password phrase is not valid.
PHRSEXPR	38	Current password phrase has expired.
SUCCESSM	40	Successful Multi-Factor Authentication
INVMFA	41	Failed Multi-Factor Authentication
MFAUNAVL	42	Multi-Factor Authentication unavailable

The Format of the ACCESS Record Extension

Table 18 on page 155 describes the format of a record that is created by the access to a resource.

Table 18. Format of the ACCESS Record Extension

Field Name	Type	Length	Position		Comments
			Start	End	
ACC_RES_NAME	Char	255	282	536	Resource name or old resource name.
ACC_REQUEST	Char	8	538	545	The access authority requested.
ACC_GRANT	Char	8	547	554	The access authority granted.
ACC_LEVEL	Integer	3	556	558	Access level of the resource.
ACC_VOL	Char	6	560	565	Volume of the resource.
ACC_OLDVOL	Char	6	567	572	OLDVOL of the resource.
ACC_CLASS	Char	8	574	581	Class name.
ACC_APPL	Char	8	583	590	Application name specified.
ACC_TYPE	Char	8	592	599	Type of resource data. Valid values are "RESOURCE" if ACC_NAME is a generic resource name, and "PROFILE" if ACC_NAME is a generic profile.
ACC_NAME	Char	246	601	846	Resource or profile name.
ACC_OWN_ID	Char	8	848	855	Name of the profile owner.
ACC_LOGSTR	Char	255	857	1111	LOGSTR= data from the RACROUTE.
ACC_RECVR	Char	8	1113	1120	User ID to which the data is directed (RECVR= on RACROUTE).
ACC_USER_NAME	Char	20	1122	1141	User name from the ACEE.
ACC_SECL	Char	8	1143	1150	SECLABEL of the resource.
ACC_UTK_ENCR	Yes/No	4	1152	1155	Is the UTOKEN associated with this user encrypted?
ACC_UTK_PRE19	Yes/No	4	1157	1160	Is this a token for a release earlier than RACF 1.9?
ACC_UTK_VERPROF	Yes/No	4	1162	1165	Is the VERIFYX propagation flag set?
ACC_UTK_NJEUNUSR	Yes/No	4	1167	1170	Is this the NJE undefined user?

Table 18. Format of the ACCESS Record Extension (continued)

Field Name	Type	Length	Position		Comments
			Start	End	
ACC_UTK_LOGUSR	Yes/No	4	1172	1175	Is UAUDIT specified for this user?
ACC_UTK_SPECIAL	Yes/No	4	1177	1180	Is this a RACF SPECIAL user?
ACC_UTK_DEFAULT	Yes/No	4	1182	1185	Is this a default token?
ACC_UTK_UNKNUSR	Yes/No	4	1187	1190	Is this an undefined user?
ACC_UTK_ERROR	Yes/No	4	1192	1195	Is this user token in error?
ACC_UTK_TRUSTED	Yes/No	4	1197	1200	Is this user a part of the trusted computing base (TCB)?
ACC_UTK_SESSTYPE	Char	8	1202	1209	The session type of this session.
ACC_UTK_SURROGAT	Yes/No	4	1211	1214	Is this a surrogate user?
ACC_UTK_REMOTE	Yes/No	4	1216	1219	Is this a remote job?
ACC_UTK_PRIV	Yes/No	4	1221	1224	Is this a privileged user ID?
ACC_UTK_SECL	Char	8	1226	1233	The SECLABEL of the user.
ACC_UTK_EXECNODE	Char	8	1235	1242	The execution node of the work.
ACC_UTK_SUSER_ID	Char	8	1244	1251	The submitting user ID.
ACC_UTK_SNODE	Char	8	1253	1260	The submitting node.
ACC_UTK_SGRP_ID	Char	8	1262	1269	The submitting group ID.
ACC_UTK_SPOE	Char	8	1271	1278	The port of entry.
ACC_UTK_SPCLASS	Char	8	1280	1287	Class of the POE. Valid values are "TERMINAL", "CONSOLE", "JESINPUT", and "APPCPORT".
ACC_UTK_USER_ID	Char	8	1289	1296	User ID associated with the record.
ACC_UTK_GRP_ID	Char	8	1298	1305	Group ID associated with the record.
ACC_UTK_DFT_GRP	Yes/No	4	1307	1310	Is a default group assigned?
ACC_UTK_DFT_SECL	Yes/No	4	1312	1315	Is a default SECLABEL assigned?
ACC_RTK_ENCR	Yes/No	4	1317	1320	Is the RTOKEN associated with this user encrypted?
ACC_RTK_PRE19	Yes/No	4	1322	1325	Is this a token for a release earlier than RACF 1.9?
ACC_RTK_VERPROF	Yes/No	4	1327	1330	Is the VERIFYX propagation flag set?
ACC_RTK_NJEUNUSR	Yes/No	4	1332	1335	Is this the NJE undefined user?
ACC_RTK_LOGUSR	Yes/No	4	1337	1340	Is UAUDIT specified for this user?
ACC_RTK_SPECIAL	Yes/No	4	1342	1345	Is this a RACF SPECIAL user?
ACC_RTK_DEFAULT	Yes/No	4	1347	1350	Is this a default token?
ACC_RTK_UNKNUSR	Yes/No	4	1352	1355	Is this an undefined user?
ACC_RTK_ERROR	Yes/No	4	1357	1360	Is this user token in error?
ACC_RTK_TRUSTED	Yes/No	4	1362	1365	Is this user a part of the trusted computing base (TCB)?
ACC_RTK_SESSTYPE	Char	8	1367	1374	The session type of this session.
ACC_RTK_SURROGAT	Yes/No	4	1376	1379	Is this a surrogate user?
ACC_RTK_REMOTE	Yes/No	4	1381	1384	Is this a remote job?
ACC_RTK_PRIV	Yes/No	4	1386	1389	Is this a privileged user ID?
ACC_RTK_SECL	Char	8	1391	1398	The SECLABEL of the user.
ACC_RTK_EXECNODE	Char	8	1400	1407	The execution node of the work.

Table 18. Format of the ACCESS Record Extension (continued)

Field Name	Type	Length	Position		Comments
			Start	End	
ACC_RTK_SUSER_ID	Char	8	1409	1416	The submitting user ID.
ACC_RTK_SNODE	Char	8	1418	1425	The submitting node.
ACC_RTK_SGRP_ID	Char	8	1427	1434	The submitting group ID.
ACC_RTK_SPOE	Char	8	1436	1443	The port of entry.
ACC_RTK_SPCLASS	Char	8	1445	1452	Class of the POE. Valid values are "TERMINAL", "CONSOLE", "JESINPUT", and "PCPORT".
ACC_RTK_USER_ID	Char	8	1454	1461	User ID associated with the record.
ACC_RTK_GRP_ID	Char	8	1463	1470	Group ID associated with the record.
ACC_RTK_DFT_GRP	Yes/No	4	1472	1475	Is a default group assigned?
ACC_RTK_DFT_SECL	Yes/No	4	1477	1480	Is a default SECLABEL assigned?
ACC_APPC_LINK	Char	16	1482	1497	A key to link together audit records for a user's APPC transaction processing work.
ACC_DCE_LINK	Char	16	1499	1514	Link to connect DCE records that originate from a single DCE request.
ACC_AUTH_TYPE	Char	13	1516	1528	Defines the type of request. Valid values are: "SERVER" and "AUTH_CLIENT" and "UNAUTH_CLIENT".

Event Qualifiers for Access Records

The event qualifiers that may be associated with an access event are shown in Table 19 on page 157.

Table 19. Event Code Qualifiers for Access Records

Event Qualifier	Event Number	Event Description
SUCCESS	00	Successful access.
INSAUTH	01	Insufficient authority.
PRFNFND	02	Profile not found; RACFIND specified on macro.
WARNING	03	Access allowed by WARNING.
FPROTALL	04	Failed by PROTECTALL.
WPROTALL	05	PROTECTALL warning.
INSCATG	06	Insufficient category or level.
INSSECL	07	Insufficient SECLABEL.
WSECLM	08	Warning: SECLABEL missing.
WINSSECL	09	Warning: Insufficient SECLABEL.
WNOTCAT	10	Warning: Data set not cataloged, but was required for authority check.
NOTCAT	11	Data set not cataloged.
PRFNFDAI	12	Profile not found.
WINSCATG	13	Warning: Insufficient category or level.

The Format of the ADDVOL Record Extension

Table 20 on page 158 describes the format of a record that is created by the ADDVOL or CHGVOL operations.

Table 20. Format of the ADDVOL Record Extension

Field Name	Type	Length	Position		Comments
			Start	End	
ADV_RES_NAME	Char	255	282	536	Resource name.
ADV_GRANT	Char	8	538	545	The access authority granted.
ADV_LEVEL	Integer	3	547	549	The level of the resource.
ADV_VOL	Char	6	551	556	Volume of the resource.
ADV_OLDVOL	Char	6	558	563	OLDVOL of the resource.
ADV_CLASS	Char	8	565	572	Class name.
ADV_OWN_ID	Char	8	574	581	Name of the profile owner.
ADV_LOGSTR	Char	255	583	837	LOGSTR= data from the RACROUTE
ADV_USER_NAME	Char	20	839	858	User name from the ACEE.
ADV_UTK_ENCR	Yes/No	4	860	863	Is the UTOKEN associated with this user encrypted?
ADV_UTK_PRE19	Yes/No	4	865	868	Is this a token for a release earlier than RACF 1.9?
ADV_UTK_VERPROF	Yes/No	4	870	873	Is the VERIFYX propagation flag set?
ADV_UTK_NJEUNUSR	Yes/No	4	875	878	Is this the NJE undefined user?
ADV_UTK_LOGUSR	Yes/No	4	880	883	Is UAUDIT specified for this user?
ADV_UTK_SPECIAL	Yes/No	4	885	888	Is this a RACF SPECIAL user?
ADV_UTK_DEFAULT	Yes/No	4	890	893	Is this a default token?
ADV_UTK_UNKNUSR	Yes/No	4	895	898	Is this an undefined user?
ADV_UTK_ERROR	Yes/No	4	900	903	Is this user token in error?
ADV_UTK_TRUSTED	Yes/No	4	905	908	Is this user a part of the trusted computing base (TCB)?
ADV_UTK_SESSTYPE	Char	8	910	917	The session type of this session.
ADV_UTK_SURROGAT	Yes/No	4	919	922	Is this a surrogate user?
ADV_UTK_REMOTE	Yes/No	4	924	927	Is this a remote job?
ADV_UTK_PRIV	Yes/No	4	929	932	Is this a privileged user ID?
ADV_UTK_SECL	Char	8	934	941	The SECLABEL of the user.
ADV_UTK_EXECNODE	Char	8	943	950	The execution node of the work.
ADV_UTK_SUSER_ID	Char	8	952	959	The submitting user ID.
ADV_UTK_SNODE	Char	8	961	968	The submitting node.
ADV_UTK_SGRP_ID	Char	8	970	977	The submitting group ID.
ADV_UTK_SPOE	Char	8	979	986	The port of entry.
ADV_UTK_SPCLASS	Char	8	988	995	Class of the POE. Valid values are "TERMINAL", "CONSOLE", "JESINPUT", and "APPCPORT".
ADV_UTK_USER_ID	Char	8	997	1004	User ID associated with the record.
ADV_UTK_GRP_ID	Char	8	1006	1013	Group ID associated with the record.
ADV_UTK_DFT_GRP	Yes/No	4	1015	1018	Is a default group assigned?
ADV_UTK_DFT_SECL	Yes/No	4	1020	1023	Is a default SECLABEL assigned?
ADV_APPC_LINK	Char	16	1025	1040	Key to link together APPC records.
ADV_SPECIFIED	Char	1024	1042	2065	The RRSF keywords specified.

Event Qualifiers for ADDVOL Records

The event qualifiers that may be associated with an ADDVOL event are shown in Table 21 on page 159.

Table 21. Event Code Qualifiers for ADDVOL Records

Event Qualifier	Event Number	Event Description
SUCCESS	00	The volume was sucessfully added or changed.
INSAUTH	01	Insufficient authority.
INSSECL	02	Insufficient SECLABEL authority.
LESSSPEC	03	A less-specific profile exists with a different SECLABEL.

The Format of the RENAMEDS Record Extension

Table 22 on page 159 describes the format of a record that is created by the rename data set, rename SFS file, or rename SFS directory operation.

Table 22. Format of the RENAMEDS Record Extension

Field Name	Type	Length	Position		Comments
			Start	End	
REN_RES_NAME	Char	255	282	536	Old resource name.
REN_NEW_RES_NAME	Char	255	538	792	New Resource name.
REN_LEVEL	Integer	3	794	796	The level of the resource.
REN_VOL	Char	6	798	803	Volume of the resource.
REN_CLASS	Char	8	805	812	Class name.
REN_OWN_ID	Char	8	814	821	Name of the profile owner.
REN_LOGSTR	Char	255	823	1077	LOGSTR= data from the RACROUTE
REN_USER_NAME	Char	20	1079	1098	User name from the ACEE.
REN_UTK_ENCR	Yes/No	4	1100	1103	Is the UTOKEN associated with this user encrypted?
REN_UTK_PRE19	Yes/No	4	1105	1108	Is this a token for a release of earlier than RACF 1.9?
REN_UTK_VERPROF	Yes/No	4	1110	1113	Is the VERIFYX propagation flag set?
REN_UTK_NJEUNUSR	Yes/No	4	1115	1118	Is this the NJE undefined user?
REN_UTK_LOGUSR	Yes/No	4	1120	1123	Is UAUDIT specified for this user?
REN_UTK_SPECIAL	Yes/No	4	1125	1128	Is this a RACF SPECIAL user?
REN_UTK_DEFAULT	Yes/No	4	1130	1133	Is this a default token?
REN_UTK_UNKNUSR	Yes/No	4	1135	1138	Is this an undefined user?
REN_UTK_ERROR	Yes/No	4	1140	1143	Is this user token in error?
REN_UTK_TRUSTED	Yes/No	4	1145	1148	Is this user a part of the trusted computing base (TCB)?
REN_UTK_SESSTYPE	Char	8	1150	1157	The session type of this session.
REN_UTK_SURROGAT	Yes/No	4	1159	1162	Is this a surrogate user?
REN_UTK_REMOTE	Yes/No	4	1164	1167	Is this a remote job?
REN_UTK_PRIV	Yes/No	4	1169	1172	Is this a privileged user ID?
REN_UTK_SECL	Char	8	1174	1181	The SECLABEL of the user.
REN_UTK_EXECNODE	Char	8	1183	1190	The execution node of the work.
REN_UTK_SUSER_ID	Char	8	1192	1199	The submitting user ID.

Table 22. Format of the RENAMEDS Record Extension (continued)

Field Name	Type	Length	Position		Comments
			Start	End	
REN_UTK_SNODE	Char	8	1201	1208	The submitting node.
REN_UTK_SGRP_ID	Char	8	1210	1217	The submitting group ID.
REN_UTK_SPOE	Char	8	1219	1226	The port of entry.
REN_UTK_SPCLASS	Char	8	1228	1235	Class of the POE. Valid values are "TERMINAL", "CONSOLE", "JESINPUT", and "APPCPORT".
REN_UTK_USER_ID	Char	8	1237	1244	User ID associated with the record.
REN_UTK_GRP_ID	Char	8	1246	1253	Group ID associated with the record.
REN_UTK_DFT_GRP	Yes/No	4	1255	1258	Is a default group assigned?
REN_UTK_DFT_SECL	Yes/No	4	1260	1263	Is a default SECLABEL assigned?
REN_APPC_LINK	Char	16	1265	1280	Key to link together APPC records.
REN_SPECIFIED	Char	1024	1282	2305	The RRSF keywords specified.

Event Qualifiers for RENAMEDS Requests

The event qualifiers that may be associated with a RENAMEDS event are shown in Table 23 on page 160.

Table 23. Event Code Qualifiers for RENAMEDS Records

Event Qualifier	Event Number	Event Description
SUCCESS	00	Successful rename.
INVGRP	01	Invalid group.
NOTINGRP	02	User not in group.
INSAUTH	03	Insufficient authority.
ALRDEFD	04	Resource already defined.
NOTRACF	05	User is not RACF-defined.
NOTPROT	06	Resource not protected.
WNOTPROT	07	Warning: Resource not protected.
NOT2RACF	08	User in second qualifier is not RACF-defined.
LESSSPEC	09	A less-specific profile exists with a different SECLABEL.
INSSECL	10	Insufficient SECLABEL authority.
RSNSECL	11	Resource not protected by SECLABEL.
NMNSECL	12	New name not protected by SECLABEL.
NODOMIN	13	New SECLABEL must dominate old SECLABEL.
WINSSECL	14	Warning: Insufficient SECLABEL authority.
WRSNSECL	15	Warning: Resource not protected by SECLABEL.
WNMNSECL	16	Warning: New name not protected by SECLABEL.
WNODOMIN	17	Warning: New SECLABEL must dominate old SECLABEL.

The Format of the DELRES Record Extension

Table 24 on page 161 describes the format of a record that is created by the delete resource operation.

Table 24. Format of the DELRES Record Extension

Field Name	Type	Length	Position		Comments
			Start	End	
DELR_RES_NAME	Char	255	282	536	Old resource name.
DELR_LEVEL	Integer	3	538	540	The level of the resource.
DELR_VOL	Char	6	542	547	Volume of the resource.
DELR_CLASS	Char	8	549	556	Class name.
DELR_OWN_ID	Char	8	558	565	Name of the profile owner.
DELR_LOGSTR	Char	255	567	821	LOGSTR= data from the RACROUTE
DELR_USER_NAME	Char	20	823	842	User name from the ACEE.
DELR_UTK_ENCR	Yes/No	4	844	847	Is the UTOKEN associated with this user encrypted?
DELR_UTK_PRE19	Yes/No	4	849	852	Is this a token for a release earlier than RACF 1.9?
DELR_UTK_VERPROF	Yes/No	4	854	857	Is the VERIFYX propagation flag set?
DELR_UTK_NJEUNUSR	Yes/No	4	859	862	Is this the NJE undefined user?
DELR_UTK_LOGUSR	Yes/No	4	864	867	Is UAUDIT specified for this user?
DELR_UTK_SPECIAL	Yes/No	4	869	872	Is this a RACF SPECIAL user?
DELR_UTK_DEFAULT	Yes/No	4	874	877	Is this a default token?
DELR_UTK_UNKNUSR	Yes/No	4	879	882	Is this an undefined user?
DELR_UTK_ERROR	Yes/No	4	884	887	Is this user token in error?
DELR_UTK_TRUSTED	Yes/No	4	889	892	Is this user a part of the trusted computing base (TCB)?
DELR_UTK_SESSTYPE	Char	8	894	901	The session type of this session.
DELR_UTK_SURROGAT	Yes/No	4	903	906	Is this a surrogate user?
DELR_UTK_REMOTE	Yes/No	4	908	911	Is this a remote job?
DELR_UTK_PRIV	Yes/No	4	913	916	Is this a privileged user ID?
DELR_UTK_SECL	Char	8	918	925	The SECLABEL of the user.
DELR_UTK_EXECNODE	Char	8	927	934	The execution node of the work.
DELR_UTK_SUSER_ID	Char	8	936	943	The submitting user ID.
DELR_UTK_SNODE	Char	8	945	952	The submitting node.
DELR_UTK_SGRP_ID	Char	8	954	961	The submitting group ID.
DELR_UTK_SPOE	Char	8	963	970	The port of entry.
DELR_UTK_SPCLASS	Char	8	972	979	Class of the POE. Valid values are "TERMINAL", "CONSOLE", "JESINPUT", and "APPCPORT".
DELR_UTK_USER_ID	Char	8	981	988	User ID associated with the record.
DELR_UTK_GRP_ID	Char	8	990	997	Group ID associated with the record.
DELR_UTK_DFT_GRP	Yes/No	4	999	1002	Is a default group assigned?
DELR_UTK_DFT_SECL	Yes/No	4	1004	1007	Is a default SECLABEL assigned?
DELR_APPC_LINK	Char	16	1009	1024	Key to link together APPC records.
DELR_SPECIFIED	Char	1024	1026	2049	The RRSF keywords specified.

Event Qualifiers for DELRES Requests

The event qualifiers that may be associated with an DELRES event are shown in Table 25 on page 162.

Table 25. Event Code Qualifiers for DELRES Records		
Event Qualifier	**Event Number**	**Event Description**
SUCCESS	00	The resource was successfully deleted.
NOTFOUND	01	Resource not found.
INVVOL	02	Invalid volume.

The Format of the DELVOL Record Extension

Table 26 on page 162 describes the format of a record that is created by the delete resource operation.

Table 26. Format of the DELVOL Record Extension					
Field Name	**Type**	**Length**	**Position**		**Comments**
			Start	**End**	
DELV_RES_NAME	Char	255	282	536	Old resource name.
DELV_LEVEL	Integer	3	538	540	The level of the resource.
DELV_VOL	Char	6	542	547	Volume of the resource.
DELV_CLASS	Char	8	549	556	Class name.
DELV_OWN_ID	Char	8	558	565	Name of the profile owner.
DELV_LOGSTR	Char	255	567	821	LOGSTR= data from the RACROUTE
DELV_USER_NAME	Char	20	823	842	User name.
DELV_UTK_ENCR	Yes/No	4	844	847	Is the UTOKEN associated with this user encrypted?
DELV_UTK_PRE19	Yes/No	4	849	852	Is this a token for a release earlier than RACF 1.9?
DELV_UTK_VERPROF	Yes/No	4	854	857	Is the VERIFYX propagation flag set?
DELV_UTK_NJEUNUSR	Yes/No	4	859	862	Is this the NJE undefined user?
DELV_UTK_LOGUSR	Yes/No	4	864	867	Is UAUDIT specified for this user?
DELV_UTK_SPECIAL	Yes/No	4	869	872	Is this a RACF SPECIAL user?
DELV_UTK_DEFAULT	Yes/No	4	874	877	Is this a default token?
DELV_UTK_UNKNUSR	Yes/No	4	879	882	Is this an undefined user?
DELV_UTK_ERROR	Yes/No	4	884	887	Is this user token in error?
DELV_UTK_TRUSTED	Yes/No	4	889	892	Is this user a part of the trusted computing base (TCB)?
DELV_UTK_SESSTYPE	Char	8	894	901	The session type of this session.
DELV_UTK_SURROGAT	Yes/No	4	903	906	Is this a surrogate user?
DELV_UTK_REMOTE	Yes/No	4	908	911	Is this a remote job?
DELV_UTK_PRIV	Yes/No	4	913	916	Is this a privileged user ID?
DELV_UTK_SECL	Char	8	918	925	The SECLABEL of the user.
DELV_UTK_EXECNODE	Char	8	927	934	The execution node of the work.
DELV_UTK_SUSER_ID	Char	8	936	943	The submitting user ID.
DELV_UTK_SNODE	Char	8	945	952	The submitting node.
DELV_UTK_SGRP_ID	Char	8	954	961	The submitting group ID.
DELV_UTK_SPOE	Char	8	963	970	The port of entry.
DELV_UTK_SPCLASS	Char	8	972	979	Class of the POE. Valid values are "TERMINAL", "CONSOLE", "JESINPUT", and "APPCPORT".
DELV_UTK_USER_ID	Char	8	981	988	User ID associated with the record.

Table 26. Format of the DELVOL Record Extension (continued)					
Field Name	**Type**	**Length**	**Position**		**Comments**
			Start	**End**	
DELV_UTK_GRP_ID	Char	8	990	997	Group ID associated with the record.
DELV_UTK_DFT_GRP	Yes/No	4	999	1002	Is a default group assigned?
DELV_UTK_DFT_SECL	Yes/No	4	1004	1007	Is a default SECLABEL assigned?
DELV_APPC_LINK	Char	16	1009	1024	Key to link together APPC records.
DELV_SPECIFIED	Char	1024	1026	2049	The RRSF keywords specified.

Event Qualifiers for DELVOL Requests

The event qualifier that may be associated with an DELVOL event is shown in Table 27 on page 163.

Table 27. Event Code Qualifiers for DELVOL Records		
Event Qualifier	**Event Number**	**Event Description**
SUCCESS	00	The volume was successfully deleted.

The Format of the DEFINE Record Extension

Table 28 on page 163 describes the format of a record that is created by the define resource operation.

Table 28. Format of the DEFINE Record Extension					
Field Name	**Type**	**Length**	**Position**		**Comments**
			Start	**End**	
DEF_RES_NAME	Char	255	282	536	Old resource name.
DEF_LEVEL	Integer	3	538	540	The level of the resource.
DEF_VOL	Char	6	542	547	Volume of the resource.
DEF_CLASS	Char	8	549	556	Class name.
DEF_MODEL_NAME	Char	255	558	812	Name of the model profile.
DEF_MODEL_VOL	Char	6	814	819	Volser of the model profile.
DEF_OWN_ID	Char	8	821	828	Owner of the profile.
DEF_LOGSTR	Char	255	830	1084	LOGSTR= data from the RACROUTE
DEF_USER_NAME	Char	20	1086	1105	User name.
DEF_UTK_ENCR	Yes/No	4	1107	1110	Is the UTOKEN associated with this user encrypted?
DEF_UTK_PRE19	Yes/No	4	1112	1115	Is this a token for a release earlier than RACF 1.9?
DEF_UTK_VERPROF	Yes/No	4	1117	1120	Is the VERIFYX propagation flag set?
DEF_UTK_NJEUNUSR	Yes/No	4	1122	1125	Is this the NJE undefined user?
DEF_UTK_LOGUSR	Yes/No	4	1127	1130	Is UAUDIT specified for this user?
DEF_UTK_SPECIAL	Yes/No	4	1132	1135	Is this a RACF SPECIAL user?
DEF_UTK_DEFAULT	Yes/No	4	1137	1140	Is this a default token?
DEF_UTK_UNKNUSR	Yes/No	4	1142	1145	Is this an undefined user?
DEF_UTK_ERROR	Yes/No	4	1147	1150	Is this user token in error?
DEF_UTK_TRUSTED	Yes/No	4	1152	1155	Is this user a part of the trusted computing base (TCB)?
DEF_UTK_SESSTYPE	Char	8	1157	1164	The session type of this session.

Table 28. Format of the DEFINE Record Extension (continued)

Field Name	Type	Length	Position		Comments
			Start	End	
DEF_UTK_SURROGAT	Yes/No	4	1166	1169	Is this a surrogate user?
DEF_UTK_REMOTE	Yes/No	4	1171	1174	Is this a remote job?
DEF_UTK_PRIV	Yes/No	4	1176	1179	Is this a privileged user ID?
DEF_UTK_SECL	Char	8	1181	1188	The SECLABEL of the user.
DEF_UTK_EXECNODE	Char	8	1190	1197	The execution node of the work.
DEF_UTK_SUSER_ID	Char	8	1199	1206	The submitting user ID.
DEF_UTK_SNODE	Char	8	1208	1215	The submitting node.
DEF_UTK_SGRP_ID	Char	8	1217	1224	The submitting group ID.
DEF_UTK_SPOE	Char	8	1226	1233	The port of entry.
DEF_UTK_SPCLASS	Char	8	1235	1242	Class of the POE. Valid values are "TERMINAL", "CONSOLE", "JESINPUT", and "APPCPORT".
DEF_UTK_USER_ID	Char	8	1244	1251	User ID associated with the record.
DEF_UTK_GRP_ID	Char	8	1253	1260	Group ID associated with the record.
DEF_UTK_DFT_GRP	Yes/No	4	1262	1265	Is a default group assigned?
DEF_UTK_DFT_SECL	Yes/No	4	1267	1270	Is a default SECLABEL assigned?
DEF_APPC_LINK	Char	16	1272	1287	Key to link together APPC records.
DEF_SPECIFIED	Char	1024	1289	2312	The RRSF keywords specified.

Event Qualifiers for DEFINE Requests

The event qualifiers that may be associated with a DEFINE event are shown in Table 29 on page 164.

Table 29. Event Code Qualifiers for DEFINE Records

Event Qualifier	Event Number	Event Description
SUCCESS	00	Successful deletion.
UNDGROUP	01	Undefined group.
USNINGRP	02	User not in group.
INSAUTH	03	Insufficient authority.
ALRDEFD	04	Resource already defined.
NOTRACF	05	User is not RACF-defined.
NOTPROT	06	Resource not protected.
WNOTPROT	07	Warning: Resource not protected.
WSECLM	08	Warning: SECLABEL missing.
WINSSECL	09	Warning: insufficient SECLABEL.
NOT2RACF	10	User in second qualifier is not RACF-defined.
INSSECL	11	Insufficient SECLABEL authority.
LESSSPEC	12	A less-specific profile exists with a different SECLABEL.

The Format of the ADDSD Record Extension

Table 30 on page 165 describes the format of a record that is created by the ADDSD command.

Table 30. Format of the ADDSD Record Extension

Field Name	Type	Length	Position		Comments
			Start	End	
AD_OWN_ID	Char	8	282	289	Owner of the profile.
AD_USER_NAME	Char	20	291	310	User name.
AD_SECL	Char	8	312	319	The SECLABEL associated with the profile.
AD_UTK_ENCR	Yes/No	4	321	324	Is the UTOKEN associated with this user encrypted?
AD_UTK_PRE19	Yes/No	4	326	329	Is this a token for a release earlier than RACF 1.9?
AD_UTK_VERPROF	Yes/No	4	331	334	Is the VERIFYX propagation flag set?
AD_UTK_NJEUNUSR	Yes/No	4	336	339	Is this the NJE undefined user?
AD_UTK_LOGUSR	Yes/No	4	341	344	Is UAUDIT specified for this user?
AD_UTK_SPECIAL	Yes/No	4	346	349	Is this a RACF SPECIAL user?
AD_UTK_DEFAULT	Yes/No	4	351	354	Is this a default token?
AD_UTK_UNKNUSR	Yes/No	4	356	359	Is this an undefined user?
AD_UTK_ERROR	Yes/No	4	361	364	Is this user token in error?
AD_UTK_TRUSTED	Yes/No	4	366	369	Is this user a part of the trusted computing base (TCB)?
AD_UTK_SESSTYPE	Char	8	371	378	The session type of this session.
AD_UTK_SURROGAT	Yes/No	4	380	383	Is this a surrogate user?
AD_UTK_REMOTE	Yes/No	4	385	388	Is this a remote job?
AD_UTK_PRIV	Yes/No	4	390	393	Is this a privileged user ID?
AD_UTK_SECL	Char	8	395	402	The SECLABEL of the user.
AD_UTK_EXECNODE	Char	8	404	411	The execution node of the work.
AD_UTK_SUSER_ID	Char	8	413	420	The submitting user ID.
AD_UTK_SNODE	Char	8	422	429	The submitting node.
AD_UTK_SGRP_ID	Char	8	431	438	The submitting group ID.
AD_UTK_SPOE	Char	8	440	447	The port of entry.
AD_UTK_SPCLASS	Char	8	449	456	Class of the POE. Valid values are "TERMINAL", "CONSOLE", "JESINPUT", and "APPCPORT".
AD_UTK_USER_ID	Char	8	458	465	User ID associated with the record.
AD_UTK_GRP_ID	Char	8	467	474	Group ID associated with the record.
AD_UTK_DFT_GRP	Yes/No	4	476	479	Is a default group assigned?
AD_UTK_DFT_SECL	Yes/No	4	481	484	Is a default SECLABEL assigned?
AD_APPC_LINK	Char	16	486	501	Key to link together APPC records.
AD_SECL_LINK	Char	16	503	518	Key to link together the data sets affected by a change of SECLABEL and the command that caused the SECLABEL change.
AD_DS_NAME	Char	44	520	563	The data set name.
AD_SPECIFIED	Char	1024	565	1588	The keywords specified.
AD_FAILED	Char	1024	1590	2613	The keywords that failed.

Event Qualifiers for ADDSD Commands

The event qualifiers that may be associated with an ADDSD command are shown in Table 31 on page 166.

Table 31. Event Code Qualifiers for ADDSD Command Records

Event Qualifier	Event Number	Event Description
SUCCESS	00	No violations detected.
INSAUTH	01	Insufficient authority.
KEYWVIOL	02	Keyword violation.
SECLSUCC	03	Successful retrieval of data set names.
SECLFAIL	04	Error during retrieval of data set names.

The Format of the ADDGROUP Record Extension

Table 32 on page 166 describes the format of a record that is created by the ADDGROUP command.

Table 32. Format of the ADDGROUP Record Extension

Field Name	Type	Length	Position		Comments
			Start	End	
AG_OWN_ID	Char	8	282	289	Owner of the profile.
AG_USER_NAME	Char	20	291	310	User name.
AG_UTK_ENCR	Yes/No	4	312	315	Is the UTOKEN associated with this user encrypted?
AG_UTK_PRE19	Yes/No	4	317	320	Is this a token for a release earlier than RACF 1.9?
AG_UTK_VERPROF	Yes/No	4	322	325	Is the VERIFYX propagation flag set?
AG_UTK_NJEUNUSR	Yes/No	4	327	330	Is this the NJE undefined user?
AG_UTK_LOGUSR	Yes/No	4	332	335	Is UAUDIT specified for this user?
AG_UTK_SPECIAL	Yes/No	4	337	340	Is this a RACF SPECIAL user?
AG_UTK_DEFAULT	Yes/No	4	342	345	Is this a default token?
AG_UTK_UNKNUSR	Yes/No	4	347	350	Is this an undefined user?
AG_UTK_ERROR	Yes/No	4	352	355	Is this user token in error?
AG_UTK_TRUSTED	Yes/No	4	357	360	Is this user a part of the trusted computing base (TCB)?
AG_UTK_SESSTYPE	Char	8	362	369	The session type of this session.
AG_UTK_SURROGAT	Yes/No	4	371	374	Is this a surrogate user?
AG_UTK_REMOTE	Yes/No	4	376	379	Is this a remote job?
AG_UTK_PRIV	Yes/No	4	381	384	Is this a privileged user ID?
AG_UTK_SECL	Char	8	386	393	The SECLABEL of the user.
AG_UTK_EXECNODE	Char	8	395	402	The execution node of the work.
AG_UTK_SUSER_ID	Char	8	404	411	The submitting user ID.
AG_UTK_SNODE	Char	8	413	420	The submitting node.
AG_UTK_SGRP_ID	Char	8	422	429	The submitting group ID.
AG_UTK_SPOE	Char	8	431	438	The port of entry.
AG_UTK_SPCLASS	Char	8	440	447	Class of the POE. Valid values are "TERMINAL", "CONSOLE", "JESINPUT", and "APPCPORT".
AG_UTK_USER_ID	Char	8	449	456	User ID associated with the record.
AG_UTK_GRP_ID	Char	8	458	465	Group ID associated with the record.
AG_UTK_DFT_GRP	Yes/No	4	467	470	Is a default group assigned?
AG_UTK_DFT_SECL	Yes/No	4	472	475	Is a default SECLABEL assigned?

Table 32. Format of the ADDGROUP Record Extension (continued)					
Field Name	**Type**	**Length**	**Position**		**Comments**
			Start	**End**	
AG_APPC_LINK	Char	16	477	492	Key to link together APPC records.
AG_GRP_ID	Char	8	494	501	The group ID.
AG_SPECIFIED	Char	1024	503	1526	The keywords specified.
AG_FAILED	Char	1024	1528	2551	The keywords that failed.

Event Qualifiers for ADDGROUP Commands

The event qualifiers that may be associated with an ADDGROUP command are shown in Table 33 on page 167.

Table 33. Event Code Qualifiers for ADDGROUP Command Records		
Event Qualifier	**Event Number**	**Event Description**
SUCCESS	00	No violations detected.
INSAUTH	01	Insufficient authority.
KEYWVIOL	02	Keyword violation.

The Format of the ADDUSER Record Extension

Table 34 on page 167 describes the format of a record that is created by the ADDUSER command.

Table 34. Format of the ADDUSER Record Extension					
Field Name	**Type**	**Length**	**Position**		**Comments**
			Start	**End**	
AU_OWN_ID	Char	8	282	289	Owner of the profile.
AU_USER_NAME	Char	20	291	310	User name.
AU_UTK_ENCR	Yes/No	4	312	315	Is the UTOKEN associated with this user encrypted?
AU_UTK_PRE19	Yes/No	4	317	320	Is this a token for a release earlier than RACF 1.9?
AU_UTK_VERPROF	Yes/No	4	322	325	Is the VERIFYX propagation flag set?
AU_UTK_NJEUNUSR	Yes/No	4	327	330	Is this the NJE undefined user?
AU_UTK_LOGUSR	Yes/No	4	332	335	Is UAUDIT specified for this user?
AU_UTK_SPECIAL	Yes/No	4	337	340	Is this a RACF SPECIAL user?
AU_UTK_DEFAULT	Yes/No	4	342	345	Is this a default token?
AU_UTK_UNKNUSR	Yes/No	4	347	350	Is this an undefined user?
AU_UTK_ERROR	Yes/No	4	352	355	Is this user token in error?
AU_UTK_TRUSTED	Yes/No	4	357	360	Is this user a part of the trusted computing base (TCB)?
AU_UTK_SESSTYPE	Char	8	362	369	The session type of this session.
AU_UTK_SURROGAT	Yes/No	4	371	374	Is this a surrogate user?
AU_UTK_REMOTE	Yes/No	4	376	379	Is this a remote job?
AU_UTK_PRIV	Yes/No	4	381	384	Is this a privileged user ID?
AU_UTK_SECL	Char	8	386	393	The SECLABEL of the user.
AU_UTK_EXECNODE	Char	8	395	402	The execution node of the work.

Table 34. Format of the ADDUSER Record Extension (continued)

Field Name	Type	Length	Position		Comments
			Start	End	
AU_UTK_SUSER_ID	Char	8	404	411	The submitting user ID.
AU_UTK_SNODE	Char	8	413	420	The submitting node.
AU_UTK_SGRP_ID	Char	8	422	429	The submitting group ID.
AU_UTK_SPOE	Char	8	431	438	The port of entry.
AU_UTK_SPCLASS	Char	8	440	447	Class of the POE. Valid values are "TERMINAL", "CONSOLE", "JESINPUT", and "APPCPORT".
AU_UTK_USER_ID	Char	8	449	456	User ID associated with the record.
AU_UTK_GRP_ID	Char	8	458	465	Group ID associated with the record.
AU_UTK_DFT_GRP	Yes/No	4	467	470	Is a default group assigned?
AU_UTK_DFT_SECL	Yes/No	4	472	475	Is a default SECLABEL assigned?
AU_APPC_LINK	Char	16	477	492	Key to link together APPC records.
AU_NOAUTH_CLAUTH	Yes/No	4	494	497	Were violations detected because the user issuing the command lacked the CLAUTH authority in the user class?
AU_NOAUTH_GROUP	Yes/No	4	499	502	Were violations detected because the user issuing the command lacked the authority within the group?
AU_USER_ID	Char	8	504	511	The user ID.
AU_SPECIFIED	Char	1024	513	1536	The keywords specified.
AU_FAILED	Char	1024	1538	2561	The keywords that failed.
AU_IGNORED	Char	1024	2563	3586	The keywords ignored.

Event Qualifiers for ADDUSER Commands

The event qualifiers that may be associated with an ADDUSER command are shown in Table 35 on page 168.

Table 35. Event Code Qualifiers for ADDUSER Command Records

Event Qualifier	Event Number	Event Description
SUCCESS	00	No violations detected.
INSAUTH	01	Insufficient authority.
KEYWVIOL	02	Keyword violation.

The Format of the ALTDSD Record Extension

Table 36 on page 168 describes the format of a record that is created by the ALTDSD command.

Table 36. Format of the ALTDSD Record Extension

Field Name	Type	Length	Position		Comments
			Start	End	
ALD_OWN_ID	Char	8	282	289	Owner of the profile.
ALD_USER_NAME	Char	20	291	310	User name.
ALD_OLD_SECL	Char	8	312	319	The SECLABEL that is being deleted from the profile.
ALD_UTK_ENCR	Yes/No	4	321	324	Is the UTOKEN associated with this user encrypted?

Table 36. Format of the ALTDSD Record Extension (continued)

Field Name	Type	Length	Position		Comments
			Start	End	
ALD_UTK_PRE19	Yes/No	4	326	329	Is this a token for a release earlier than RACF 1.9?
ALD_UTK_VERPROF	Yes/No	4	331	334	Is the VERIFYX propagation flag set?
ALD_UTK_NJEUNUSR	Yes/No	4	336	339	Is this the NJE undefined user?
ALD_UTK_LOGUSR	Yes/No	4	341	344	Is UAUDIT specified for this user?
ALD_UTK_SPECIAL	Yes/No	4	346	349	Is this a RACF SPECIAL user?
ALD_UTK_DEFAULT	Yes/No	4	351	354	Is this a default token?
ALD_UTK_UNKNUSR	Yes/No	4	356	359	Is this an undefined user?
ALD_UTK_ERROR	Yes/No	4	361	364	Is this user token in error?
ALD_UTK_TRUSTED	Yes/No	4	366	369	Is this user a part of the trusted computing base (TCB)?
ALD_UTK_SESSTYPE	Char	8	371	378	The session type of this session.
ALD_UTK_SURROGAT	Yes/No	4	380	383	Is this a surrogate user?
ALD_UTK_REMOTE	Yes/No	4	385	388	Is this a remote job?
ALD_UTK_PRIV	Yes/No	4	390	393	Is this a privileged user ID?
ALD_UTK_SECL	Char	8	395	402	The SECLABEL of the user.
ALD_UTK_EXECNODE	Char	8	404	411	The execution node of the work.
ALD_UTK_SUSER_ID	Char	8	413	420	The submitting user ID.
ALD_UTK_SNODE	Char	8	422	429	The submitting node.
ALD_UTK_SGRP_ID	Char	8	431	438	The submitting group ID.
ALD_UTK_SPOE	Char	8	440	447	The port of entry.
ALD_UTK_SPCLASS	Char	8	449	456	Class of the POE. Valid values are "TERMINAL", "CONSOLE", "JESINPUT", and "APPCPORT".
ALD_UTK_USER_ID	Char	8	458	465	User ID associated with the record.
ALD_UTK_GRP_ID	Char	8	467	474	Group ID associated with the record.
ALD_UTK_DFT_GRP	Yes/No	4	476	479	Is a default group assigned?
ALD_UTK_DFT_SECL	Yes/No	4	481	484	Is a default SECLABEL assigned?
ALD_APPC_LINK	Char	16	486	501	Key to link together APPC records.
ALD_SECL_LINK	Char	16	503	518	Key to link together the data sets affected by a change of SECLABEL and the command that caused the SECLABEL change.
ALD_DS_NAME	Char	44	520	563	The data set name.
ALD_SPECIFIED	Char	1024	565	1588	The keywords specified.
ALD_FAILED	Char	1024	1590	2613	The keywords that failed.
ALD_IGNORED	Char	1024	2615	3638	The keywords ignored.

Event Qualifiers for ALTDSD Commands

The event qualifiers that may be associated with an ALTDSD command are shown in Table 37 on page 170.

Table 37. Event Code Qualifiers for ALTDSD Command Records

Event Qualifier	Event Number	Event Description
SUCCESS	00	No violations detected.
INSAUTH	01	Insufficient authority.
KEYWVIOL	02	Keyword violation.
SECLSUCC	03	Successful retrieval of data set names.
SECLFAIL	04	Error during retrieval of data set names.

The Format of the ALTGROUP Record Extension

Table 38 on page 170 describes the format of a record that is created by the ALTGROUP command.

Table 38. Format of the ALTGROUP Record Extension

Field Name	Type	Length	Position		Comments
			Start	End	
ALG_OWN_ID	Char	8	282	289	Owner of the profile.
ALG_USER_NAME	Char	20	291	310	User name.
ALG_UTK_ENCR	Yes/No	4	312	315	Is the UTOKEN associated with this user encrypted?
ALG_UTK_PRE19	Yes/No	4	317	320	Is this a token for a release earlier than RACF 1.9?
ALG_UTK_VERPROF	Yes/No	4	322	325	Is the VERIFYX propagation flag set?
ALG_UTK_NJEUNUSR	Yes/No	4	327	330	Is this the NJE undefined user?
ALG_UTK_LOGUSR	Yes/No	4	332	335	Is UAUDIT specified for this user?
ALG_UTK_SPECIAL	Yes/No	4	337	340	Is this a RACF SPECIAL user?
ALG_UTK_DEFAULT	Yes/No	4	342	345	Is this a default token?
ALG_UTK_UNKNUSR	Yes/No	4	347	350	Is this an undefined user?
ALG_UTK_ERROR	Yes/No	4	352	355	Is this user token in error?
ALG_UTK_TRUSTED	Yes/No	4	357	360	Is this user a part of the trusted computing base (TCB)?
ALG_UTK_SESSTYPE	Char	8	362	369	The session type of this session.
ALG_UTK_SURROGAT	Yes/No	4	371	374	Is this a surrogate user?
ALG_UTK_REMOTE	Yes/No	4	376	379	Is this a remote job?
ALG_UTK_PRIV	Yes/No	4	381	384	Is this a privileged user ID?
ALG_UTK_SECL	Char	8	386	393	The SECLABEL of the user.
ALG_UTK_EXECNODE	Char	8	395	402	The execution node of the work.
ALG_UTK_SUSER_ID	Char	8	404	411	The submitting user ID.
ALG_UTK_SNODE	Char	8	413	420	The submitting node.
ALG_UTK_SGRP_ID	Char	8	422	429	The submitting group ID.
ALG_UTK_SPOE	Char	8	431	438	The port of entry.
ALG_UTK_SPCLASS	Char	8	440	447	Class of the POE. Valid values are "TERMINAL", "CONSOLE", "JESINPUT", and "APPCPORT".
ALG_UTK_USER_ID	Char	8	449	456	User ID associated with the record.
ALG_UTK_GRP_ID	Char	8	458	465	Group ID associated with the record.
ALG_UTK_DFT_GRP	Yes/No	4	467	470	Is a default group assigned?
ALG_UTK_DFT_SECL	Yes/No	4	472	475	Is a default SECLABEL assigned?

Table 38. Format of the ALTGROUP Record Extension (continued)

Field Name	Type	Length	Position		Comments
			Start	End	
ALG_APPC_LINK	Char	16	477	492	Key to link together APPC records.
ALG_GRP_ID	Char	8	494	501	The group ID.
ALG_SPECIFIED	Char	1024	503	1526	The keywords specified.
ALG_FAILED	Char	1024	1528	2551	The keywords that failed.
ALG_IGNORED	Char	1024	2553	3576	The keywords ignored.

Event Qualifiers for ALTGROUP Commands

The event qualifiers that may be associated with an ALTGROUP command are shown in Table 39 on page 171.

Table 39. Event Code Qualifiers for ALTGROUP Command Records

Event Qualifier	Event Number	Event Description
SUCCESS	00	No violations detected.
INSAUTH	01	Insufficient authority.
KEYWVIOL	02	Keyword violation.

The Format of the ALTUSER Record Extension

Table 40 on page 171 describes the format of a record that is created by the ALTUSER command.

Table 40. Format of the ALTUSER Record Extension

Field Name	Type	Length	Position		Comments
			Start	End	
ALU_OWN_ID	Char	8	282	289	Owner of the profile.
ALU_USER_NAME	Char	20	291	310	User name.
ALU_OLD_SECL	Char	8	312	319	The SECLABEL that is being deleted from the profile.
ALU_UTK_ENCR	Yes/No	4	321	324	Is the UTOKEN associated with this user encrypted?
ALU_UTK_PRE19	Yes/No	4	326	329	Is this a token for a release earlier than RACF 1.9?
ALU_UTK_VERPROF	Yes/No	4	331	334	Is the VERIFYX propagation flag set?
ALU_UTK_NJEUNUSR	Yes/No	4	336	339	Is this the NJE undefined user?
ALU_UTK_LOGUSR	Yes/No	4	341	344	Is UAUDIT specified for this user?
ALU_UTK_SPECIAL	Yes/No	4	346	349	Is this a RACF SPECIAL user?
ALU_UTK_DEFAULT	Yes/No	4	351	354	Is this a default token?
ALU_UTK_UNKNUSR	Yes/No	4	356	359	Is this an undefined user?
ALU_UTK_ERROR	Yes/No	4	361	364	Is this user token in error?
ALU_UTK_TRUSTED	Yes/No	4	366	369	Is this user a part of the trusted computing base (TCB)?
ALU_UTK_SESSTYPE	Char	8	371	378	The session type of this session.
ALU_UTK_SURROGAT	Yes/No	4	380	383	Is this a surrogate user?
ALU_UTK_REMOTE	Yes/No	4	385	388	Is this a remote job?
ALU_UTK_PRIV	Yes/No	4	390	393	Is this a privileged user ID?

Table 40. Format of the ALTUSER Record Extension (continued)

Field Name	Type	Length	Position		Comments
			Start	End	
ALU_UTK_SECL	Char	8	395	402	The SECLABEL of the user.
ALU_UTK_EXECNODE	Char	8	404	411	The execution node of the work.
ALU_UTK_SUSER_ID	Char	8	413	420	The submitting user ID.
ALU_UTK_SNODE	Char	8	422	429	The submitting node.
ALU_UTK_SGRP_ID	Char	8	431	438	The submitting group ID.
ALU_UTK_SPOE	Char	8	440	447	The port of entry.
ALU_UTK_SPCLASS	Char	8	449	456	Class of the POE. Valid values are "TERMINAL", "CONSOLE", "JESINPUT", and "APPCPORT".
ALU_UTK_USER_ID	Char	8	458	465	User ID associated with the record.
ALU_UTK_GRP_ID	Char	8	467	474	Group ID associated with the record.
ALU_UTK_DFT_GRP	Yes/No	4	476	479	Is a default group assigned?
ALU_UTK_DFT_SECL	Yes/No	4	481	484	Is a default SECLABEL assigned?
ALU_APPC_LINK	Char	16	486	501	Key to link together APPC records.
ALU_NOAUTH_CLAUTH	Yes/No	4	503	506	Were violations detected because the user issuing the command lacked the CLAUTH authority in the user class?
ALU_NOAUTH_GROUP	Yes/No	4	508	511	Were violations detected because the user issuing the command lacked the authority within the group?
ALU_NOAUTH_PROF	Yes/No	4	513	516	Were violations detected because the user issuing the command lacked authority to the profile?
ALU_USER_ID	Char	8	518	525	The user ID.
ALU_SPECIFIED	Char	1024	527	1550	The keywords specified.
ALU_FAILED	Char	1024	1552	2575	The keywords that failed.
ALU_IGNORED	Char	1024	2577	3600	The keywords ignored.

Event Qualifiers for ALTUSER Commands

The event qualifiers that may be associated with an ALTUSER command are shown in Table 41 on page 172.

Table 41. Event Code Qualifiers for ALTUSER Command Records

Event Qualifier	Event Number	Event Description
SUCCESS	00	No violations detected.
INSAUTH	01	Insufficient authority.
KEYWVIOL	02	Keyword violation.

The Format of the CONNECT Record Extension

Table 42 on page 172 describes the format of a record that is created by the CONNECT command.

Table 42. Format of the CONNECT Record Extension

Field Name	Type	Length	Position		Comments
			Start	End	
CON_OWN_ID	Char	8	282	289	Owner of the profile.

Table 42. Format of the CONNECT Record Extension (continued)

Field Name	Type	Length	Position		Comments
			Start	End	
CON_USER_NAME	Char	20	291	310	User name.
CON_UTK_ENCR	Yes/No	4	312	315	Is the UTOKEN associated with this user encrypted?
CON_UTK_PRE19	Yes/No	4	317	320	Is this a token for a release earlier than RACF 1.9?
CON_UTK_VERPROF	Yes/No	4	322	325	Is the VERIFYX propagation flag set?
CON_UTK_NJEUNUSR	Yes/No	4	327	330	Is this the NJE undefined user?
CON_UTK_LOGUSR	Yes/No	4	332	335	Is UAUDIT specified for this user?
CON_UTK_SPECIAL	Yes/No	4	337	340	Is this a RACF SPECIAL user?
CON_UTK_DEFAULT	Yes/No	4	342	345	Is this a default token?
CON_UTK_UNKNUSR	Yes/No	4	347	350	Is this an undefined user?
CON_UTK_ERROR	Yes/No	4	352	355	Is this user token in error?
CON_UTK_TRUSTED	Yes/No	4	357	360	Is this user a part of the trusted computing base (TCB)?
CON_UTK_SESSTYPE	Char	8	362	369	The session type of this session.
CON_UTK_SURROGAT	Yes/No	4	371	374	Is this a surrogate user?
CON_UTK_REMOTE	Yes/No	4	376	379	Is this a remote job?
CON_UTK_PRIV	Yes/No	4	381	384	Is this a privileged user ID?
CON_UTK_SECL	Char	8	386	393	The SECLABEL of the user.
CON_UTK_EXECNODE	Char	8	395	402	The execution node of the work.
CON_UTK_SUSER_ID	Char	8	404	411	The submitting user ID.
CON_UTK_SNODE	Char	8	413	420	The submitting node.
CON_UTK_SGRP_ID	Char	8	422	429	The submitting group ID.
CON_UTK_SPOE	Char	8	431	438	The port of entry.
CON_UTK_SPCLASS	Char	8	440	447	Class of the POE. Valid values are "TERMINAL", "CONSOLE", "JESINPUT", and "APPCPORT".
CON_UTK_USER_ID	Char	8	449	456	User ID associated with the record.
CON_UTK_GRP_ID	Char	8	458	465	Group ID associated with the record.
CON_UTK_DFT_GRP	Yes/No	4	467	470	Is a default group assigned?
CON_UTK_DFT_SECL	Yes/No	4	472	475	Is a default SECLABEL assigned?
CON_APPC_LINK	Char	16	477	492	Key to link together APPC records.
CON_USER_ID	Char	8	494	501	The user ID that is being connected.
CON_SPECIFIED	Char	1024	503	1526	The keywords specified.
CON_FAILED	Char	1024	1528	2551	The keywords that failed.

Event Qualifiers for CONNECT Commands

The event qualifiers that may be associated with an CONNECT command are shown in Table 43 on page 173.

Table 43. Event Code Qualifiers for CONNECT Command Records

Event Qualifier	Event Number	Event Description
SUCCESS	00	No violations detected.

Table 43. Event Code Qualifiers for CONNECT Command Records (continued)		
Event Qualifier	**Event Number**	**Event Description**
INSAUTH	01	Insufficient authority.
KEYWVIOL	02	Keyword violation.

The Format of the DELDSD Record Extension

Table 44 on page 174 describes the format of a record that is created by the DELDSD command.

Table 44. Format of the DELDSD Record Extension

Field Name	Type	Length	Position		Comments
			Start	End	
DELD_OWN_ID	Char	8	282	289	Owner of the profile.
DELD_USER_NAME	Char	20	291	310	User name.
DELD_OLD_SECL	Char	8	312	319	The SECLABEL that is being deleted.
DELD_UTK_ENCR	Yes/No	4	321	324	Is the UTOKEN associated with this user encrypted?
DELD_UTK_PRE19	Yes/No	4	326	329	Is this a token for a release earlier than RACF 1.9?
DELD_UTK_VERPROF	Yes/No	4	331	334	Is the VERIFYX propagation flag set?
DELD_UTK_NJEUNUSR	Yes/No	4	336	339	Is this the NJE undefined user?
DELD_UTK_LOGUSR	Yes/No	4	341	344	Is UAUDIT specified for this user?
DELD_UTK_SPECIAL	Yes/No	4	346	349	Is this a RACF SPECIAL user?
DELD_UTK_DEFAULT	Yes/No	4	351	354	Is this a default token?
DELD_UTK_UNKNUSR	Yes/No	4	356	359	Is this an undefined user?
DELD_UTK_ERROR	Yes/No	4	361	364	Is this user token in error?
DELD_UTK_TRUSTED	Yes/No	4	366	369	Is this user a part of the trusted computing base (TCB)?
DELD_UTK_SESSTYPE	Char	8	371	378	The session type of this session.
DELD_UTK_SURROGAT	Yes/No	4	380	383	Is this a surrogate user?
DELD_UTK_REMOTE	Yes/No	4	385	388	Is this a remote job?
DELD_UTK_PRIV	Yes/No	4	390	393	Is this a privileged user ID?
DELD_UTK_SECL	Char	8	395	402	The SECLABEL of the user.
DELD_UTK_EXECNODE	Char	8	404	411	The execution node of the work.
DELD_UTK_SUSER_ID	Char	8	413	420	The submitting user ID.
DELD_UTK_SNODE	Char	8	422	429	The submitting node.
DELD_UTK_SGRP_ID	Char	8	431	438	The submitting group ID.
DELD_UTK_SPOE	Char	8	440	447	The port of entry.
DELD_UTK_SPCLASS	Char	8	449	456	Class of the POE. Valid values are "TERMINAL", "CONSOLE", "JESINPUT", and "APPCPORT".
DELD_UTK_USER_ID	Char	8	458	465	User ID associated with the record.
DELD_UTK_GRP_ID	Char	8	467	474	Group ID associated with the record.
DELD_UTK_DFT_GRP	Yes/No	4	476	479	Is a default group assigned?
DELD_UTK_DFT_SECL	Yes/No	4	481	484	Is a default SECLABEL assigned?
DELD_APPC_LINK	Char	16	486	501	Key to link together APPC records.

Table 44. Format of the DELDSD Record Extension (continued)

Field Name	Type	Length	Position		Comments
			Start	End	
DELD_SECL_LINK	Char	16	503	518	Key to link together the data sets affected by a change of SECLABEL and the command that caused the SECLABEL change.
DELD_DS_NAME	Char	44	520	563	The data set profile that is being deleted.
DELD_SPECIFIED	Char	1024	565	1588	The keywords specified.
DELD_FAILED	Char	1024	1590	2613	The keywords that failed.

Event Qualifiers for DELDSD Commands

The event qualifiers that may be associated with an DELDSD command are shown in Table 45 on page 175.

Table 45. Event Code Qualifiers for DELDSD Command Records

Event Qualifier	Event Number	Event Description
SUCCESS	00	No violations detected.
INSAUTH	01	Insufficient authority.
KEYWVIOL	02	Keyword violation.
SECLSUCC	03	Successful retrieval of data set names.
SECLFAIL	04	Error during retrieval of data set names.

The Format of the DELGROUP Record Extension

Table 46 on page 175 describes the format of a record that is created by the DELGROUP command.

Table 46. Format of the DELGROUP Record Extension

Field Name	Type	Length	Position		Comments
			Start	End	
DELG_OWN_ID	Char	8	282	289	Owner of the profile.
DELG_USER_NAME	Char	20	291	310	User name.
DELG_UTK_ENCR	Yes/No	4	312	315	Is the UTOKEN associated with this user encrypted?
DELG_UTK_PRE19	Yes/No	4	317	320	Is this a token for a release earlier than RACF 1.9?
DELG_UTK_VERPROF	Yes/No	4	322	325	Is the VERIFYX propagation flag set?
DELG_UTK_NJEUNUSR	Yes/No	4	327	330	Is this the NJE undefined user?
DELG_UTK_LOGUSR	Yes/No	4	332	335	Is UAUDIT specified for this user?
DELG_UTK_SPECIAL	Yes/No	4	337	340	Is this a RACF SPECIAL user?
DELG_UTK_DEFAULT	Yes/No	4	342	345	Is this a default token?
DELG_UTK_UNKNUSR	Yes/No	4	347	350	Is this an undefined user?
DELG_UTK_ERROR	Yes/No	4	352	355	Is this user token in error?
DELG_UTK_TRUSTED	Yes/No	4	357	360	Is this user a part of the trusted computing base (TCB)?
DELG_UTK_SESSTYPE	Char	8	362	369	The session type of this session.
DELG_UTK_SURROGAT	Yes/No	4	371	374	Is this a surrogate user?
DELG_UTK_REMOTE	Yes/No	4	376	379	Is this a remote job?

Table 46. Format of the DELGROUP Record Extension (continued)

Field Name	Type	Length	Position		Comments
			Start	End	
DELG_UTK_PRIV	Yes/No	4	381	384	Is this a privileged user ID?
DELG_UTK_SECL	Char	8	386	393	The SECLABEL of the user.
DELG_UTK_EXECNODE	Char	8	395	402	The execution node of the work.
DELG_UTK_SUSER_ID	Char	8	404	411	The submitting user ID.
DELG_UTK_SNODE	Char	8	413	420	The submitting node.
DELG_UTK_SGRP_ID	Char	8	422	429	The submitting group ID.
DELG_UTK_SPOE	Char	8	431	438	The port of entry.
DELG_UTK_SPCLASS	Char	8	440	447	Class of the POE. Valid values are "TERMINAL", "CONSOLE", "JESINPUT", and "APPCPORT".
DELG_UTK_USER_ID	Char	8	449	456	User ID associated with the record.
DELG_UTK_GRP_ID	Char	8	458	465	Group ID associated with the record.
DELG_UTK_DFT_GRP	Yes/No	4	467	470	Is a default group assigned?
DELG_UTK_DFT_SECL	Yes/No	4	472	475	Is a default SECLABEL assigned?
DELG_APPC_LINK	Char	16	477	492	Key to link together APPC records.
DELG_GRP_ID	Char	8	494	501	The group that is being deleted.
DELG_SPECIFIED	Char	1024	503	1526	The keywords specified.

Event Qualifiers for DELGROUP Commands

The event qualifiers that may be associated with an DELGROUP command are shown in Table 47 on page 176.

Table 47. Event Code Qualifiers for DELGROUP Command Records

Event Qualifier	Event Number	Event Description
SUCCESS	00	No violations detected.
INSAUTH	01	Insufficient authority.
KEYWVIOL	02	Keyword violation.

The Format of the DELUSER Record Extension

Table 48 on page 176 describes the format of a record that is created by the DELUSER command.

Table 48. Format of the DELUSER Record Extension

Field Name	Type	Length	Position		Comments
			Start	End	
DELU_OWN_ID	Char	8	282	289	Owner of the profile.
DELU_USER_NAME	Char	20	291	310	User name.
DELU_UTK_ENCR	Yes/No	4	312	315	Is the UTOKEN associated with this user encrypted?
DELU_UTK_PRE19	Yes/No	4	317	320	Is this a token for a release earlier than RACF 1.9?
DELU_UTK_VERPROF	Yes/No	4	322	325	Is the VERIFYX propagation flag set?
DELU_UTK_NJEUNUSR	Yes/No	4	327	330	Is this the NJE undefined user?
DELU_UTK_LOGUSR	Yes/No	4	332	335	Is UAUDIT specified for this user?

Table 48. Format of the DELUSER Record Extension (continued)

Field Name	Type	Length	Position		Comments
			Start	End	
DELU_UTK_SPECIAL	Yes/No	4	337	340	Is this a RACF SPECIAL user?
DELU_UTK_DEFAULT	Yes/No	4	342	345	Is this a default token?
DELU_UTK_UNKNUSR	Yes/No	4	347	350	Is this an undefined user?
DELU_UTK_ERROR	Yes/No	4	352	355	Is this user token in error?
DELU_UTK_TRUSTED	Yes/No	4	357	360	Is this user a part of the trusted computing base (TCB)?
DELU_UTK_SESSTYPE	Char	8	362	369	The session type of this session.
DELU_UTK_SURROGAT	Yes/No	4	371	374	Is this a surrogate user?
DELU_UTK_REMOTE	Yes/No	4	376	379	Is this a remote job?
DELU_UTK_PRIV	Yes/No	4	381	384	Is this a privileged user ID?
DELU_UTK_SECL	Char	8	386	393	The SECLABEL of the user.
DELU_UTK_EXECNODE	Char	8	395	402	The execution node of the work.
DELU_UTK_SUSER_ID	Char	8	404	411	The submitting user ID.
DELU_UTK_SNODE	Char	8	413	420	The submitting node.
DELU_UTK_SGRP_ID	Char	8	422	429	The submitting group ID.
DELU_UTK_SPOE	Char	8	431	438	The port of entry.
DELU_UTK_SPCLASS	Char	8	440	447	Class of the POE. Valid values are "TERMINAL", "CONSOLE", "JESINPUT", and "APPCPORT".
DELU_UTK_USER_ID	Char	8	449	456	User ID associated with the record.
DELU_UTK_GRP_ID	Char	8	458	465	Group ID associated with the record.
DELU_UTK_DFT_GRP	Yes/No	4	467	470	Is a default group assigned?
DELU_UTK_DFT_SECL	Yes/No	4	472	475	Is a default SECLABEL assigned?
DELU_APPC_LINK	Char	16	477	492	Key to link together APPC records.
DELU_USER_ID	Char	8	494	501	The user ID that is being deleted.
DELU_SPECIFIED	Char	1024	503	1526	The keywords specified.

Event Qualifiers for DELUSER Commands

The event qualifiers that may be associated with a DELUSER command are shown in Table 49 on page 177.

Table 49. Event Code Qualifiers for DELUSER Command Records

Event Qualifier	Event Number	Event Description
SUCCESS	00	No violations detected.
INSAUTH	01	Insufficient authority.
KEYWVIOL	02	Keyword violation.

The Format of the PASSWORD Record Extension

Table 50 on page 178 describes the format of a record that is created by the PASSWORD command.

Table 50. Format of the PASSWORD Record Extension

Field Name	Type	Length	Position		Comments
			Start	End	
PWD_OWN_ID	Char	8	282	289	Owner of the profile.
PWD_USER_NAME	Char	20	291	310	User name.
PWD_UTK_ENCR	Yes/No	4	312	315	Is the UTOKEN associated with this user encrypted?
PWD_UTK_PRE19	Yes/No	4	317	320	Is this a token for a release earlier than RACF 1.9?
PWD_UTK_VERPROF	Yes/No	4	322	325	Is the VERIFYX propagation flag set?
PWD_UTK_NJEUNUSR	Yes/No	4	327	330	Is this the NJE undefined user?
PWD_UTK_LOGUSR	Yes/No	4	332	335	Is UAUDIT specified for this user?
PWD_UTK_SPECIAL	Yes/No	4	337	340	Is this a RACF SPECIAL user?
PWD_UTK_DEFAULT	Yes/No	4	342	345	Is this a default token?
PWD_UTK_UNKNUSR	Yes/No	4	347	350	Is this an undefined user?
PWD_UTK_ERROR	Yes/No	4	352	355	Is this user token in error?
PWD_UTK_TRUSTED	Yes/No	4	357	360	Is this user a part of the trusted computing base (TCB)?
PWD_UTK_SESSTYPE	Char	8	362	369	The session type of this session.
PWD_UTK_SURROGAT	Yes/No	4	371	374	Is this a surrogate user?
PWD_UTK_REMOTE	Yes/No	4	376	379	Is this a remote job?
PWD_UTK_PRIV	Yes/No	4	381	384	Is this a privileged user ID?
PWD_UTK_SECL	Char	8	386	393	The SECLABEL of the user.
PWD_UTK_EXECNODE	Char	8	395	402	The execution node of the work.
PWD_UTK_SUSER_ID	Char	8	404	411	The submitting user ID.
PWD_UTK_SNODE	Char	8	413	420	The submitting node.
PWD_UTK_SGRP_ID	Char	8	422	429	The submitting group ID.
PWD_UTK_SPOE	Char	8	431	438	The port of entry.
PWD_UTK_SPCLASS	Char	8	440	447	Class of the POE. Valid values are "TERMINAL", "CONSOLE", "JESINPUT", and "APPCPORT".
PWD_UTK_USER_ID	Char	8	449	456	User ID associated with the record.
PWD_UTK_GRP_ID	Char	8	458	465	Group ID associated with the record.
PWD_UTK_DFT_GRP	Yes/No	4	467	470	Is a default group assigned?
PWD_UTK_DFT_SECL	Yes/No	4	472	475	Is a default SECLABEL assigned?
PWD_APPC_LINK	Char	16	477	492	A key to link together audit records for a user's APPC transaction processing work.
PWD_SPECIFIED	Char	1024	494	1517	The keywords specified.
PWD_FAILED	Char	1024	1519	2542	The keywords that failed.
PWD_IGNORED	Char	1024	2544	3567	The keywords ignored.

Event Qualifiers for PASSWORD Commands

The event qualifiers that may be associated with a PASSWORD command are shown in Table 51 on page 179.

Table 51. Event Code Qualifiers for PASSWORD Command Records		
Event Qualifier	**Event Number**	**Event Description**
SUCCESS	00	No violations detected.
INSAUTH	01	Insufficient authority.
KEYWVIOL	02	Keyword violation.

The Format of the PERMIT Record Extension

Table 52 on page 179 describes the format of a record that is created by the PERMIT, PERMDIR, or PERMFILE command.

Table 52. Format of the PERMIT Record Extension

Field Name	Type	Length	Position		Comments
			Start	End	
PERM_CLASS	Char	8	282	289	Class name.
PERM_OWN_ID	Char	8	291	298	Owner of the profile.
PERM_USER_NAME	Char	20	300	319	User name.
PERM_UTK_ENCR	Yes/No	4	321	324	Is the UTOKEN associated with this user encrypted?
PERM_UTK_PRE19	Yes/No	4	326	329	Is this a token for a release earlier than RACF 1.9?
PERM_UTK_VERPROF	Yes/No	4	331	334	Is the VERIFYX propagation flag set?
PERM_UTK_NJEUNUSR	Yes/No	4	336	339	Is this the NJE undefined user?
PERM_UTK_LOGUSR	Yes/No	4	341	344	Is UAUDIT specified for this user?
PERM_UTK_SPECIAL	Yes/No	4	346	349	Is this a RACF SPECIAL user?
PERM_UTK_DEFAULT	Yes/No	4	351	354	Is this a default token?
PERM_UTK_UNKNUSR	Yes/No	4	356	359	Is this an undefined user?
PERM_UTK_ERROR	Yes/No	4	361	364	Is this user token in error?
PERM_UTK_TRUSTED	Yes/No	4	366	369	Is this user a part of the trusted computing base (TCB)?
PERM_UTK_SESSTYPE	Char	8	371	378	The session type of this session.
PERM_UTK_SURROGAT	Yes/No	4	380	383	Is this a surrogate user?
PERM_UTK_REMOTE	Yes/No	4	385	388	Is this a remote job?
PERM_UTK_PRIV	Yes/No	4	390	393	Is this a privileged user ID?
PERM_UTK_SECL	Char	8	395	402	The SECLABEL of the user.
PERM_UTK_EXECNODE	Char	8	404	411	The execution node of the work.
PERM_UTK_SUSER_ID	Char	8	413	420	The submitting user ID.
PERM_UTK_SNODE	Char	8	422	429	The submitting node.
PERM_UTK_SGRP_ID	Char	8	431	438	The submitting group ID.
PERM_UTK_SPOE	Char	8	440	447	The port of entry.
PERM_UTK_SPCLASS	Char	8	449	456	Class of the POE. Valid values are "TERMINAL", "CONSOLE", "JESINPUT", and "APPCPORT".
PERM_UTK_USER_ID	Char	8	458	465	User ID associated with the record.
PERM_UTK_GRP_ID	Char	8	467	474	Group ID associated with the record.
PERM_UTK_DFT_GRP	Yes/No	4	476	479	Is a default group assigned?
PERM_UTK_DFT_SECL	Yes/No	4	481	484	Is a default SECLABEL assigned?

Table 52. Format of the PERMIT Record Extension (continued)

Field Name	Type	Length	Position		Comments
			Start	End	
PERM_APPC_LINK	Char	16	486	501	Key to link together APPC records.
PERM_RES_NAME	Char	255	503	757	The resource name
PERM_SPECIFIED	Char	1024	759	1782	The keywords specified.
PERM_FAILED	Char	1024	1784	2807	The keywords that failed.
PERM_IGNORED	Char	1024	2809	3832	The keywords ignored.

Event Qualifiers for PERMIT Commands

The event qualifiers that may be associated with a PERMIT, PERMDIR, or PERMFILE command are shown in Table 53 on page 180.

Table 53. Event Code Qualifiers for PERMIT Command Records

Event Qualifier	Event Number	Event Description
SUCCESS	00	No violations detected.
INSAUTH	01	Insufficient authority.
KEYWVIOL	02	Keyword violation.

The Format of the RALTER Record Extension

Table 54 on page 180 describes the format of a record that is created by the RALTER, ALTDIR, or ALTFILE command.

Table 54. Format of the RALTER Record Extension

Field Name	Type	Length	Position		Comments
			Start	End	
RALT_CLASS	Char	8	282	289	Class name.
RALT_OWN_ID	Char	8	291	298	Owner of the profile.
RALT_USER_NAME	Char	20	300	319	User name.
RALT_OLD_SECL	Char	8	321	328	The SECLABEL being deleted from the profile.
RALT_UTK_ENCR	Yes/No	4	330	333	Is the UTOKEN associated with this user encrypted?
RALT_UTK_PRE19	Yes/No	4	335	338	Is this a token for a release earlier than RACF 1.9?
RALT_UTK_VERPROF	Yes/No	4	340	343	Is the VERIFYX propagation flag set?
RALT_UTK_NJEUNUSR	Yes/No	4	345	348	Is this the NJE undefined user?
RALT_UTK_LOGUSR	Yes/No	4	350	353	Is UAUDIT specified for this user?
RALT_UTK_SPECIAL	Yes/No	4	355	358	Is this a RACF SPECIAL user?
RALT_UTK_DEFAULT	Yes/No	4	360	363	Is this a default token?
RALT_UTK_UNKNUSR	Yes/No	4	365	368	Is this an undefined user?
RALT_UTK_ERROR	Yes/No	4	370	373	Is this user token in error?
RALT_UTK_TRUSTED	Yes/No	4	375	378	Is this user a part of the trusted computing base (TCB)?
RALT_UTK_SESSTYPE	Char	8	380	387	The session type of this session.
RALT_UTK_SURROGAT	Yes/No	4	389	392	Is this a surrogate user?

Table 54. Format of the RALTER Record Extension (continued)

Field Name	Type	Length	Position		Comments
			Start	End	
RALT_UTK_REMOTE	Yes/No	4	394	397	Is this a remote job?
RALT_UTK_PRIV	Yes/No	4	399	402	Is this a privileged user ID?
RALT_UTK_SECL	Char	8	404	411	The SECLABEL of the user.
RALT_UTK_EXECNODE	Char	8	413	420	The execution node of the work.
RALT_UTK_SUSER_ID	Char	8	422	429	The submitting user ID.
RALT_UTK_SNODE	Char	8	431	438	The submitting node.
RALT_UTK_SGRP_ID	Char	8	440	447	The submitting group ID.
RALT_UTK_SPOE	Char	8	449	456	The port of entry.
RALT_UTK_SPCLASS	Char	8	458	465	Class of the POE. Valid values are "TERMINAL", "CONSOLE", "JESINPUT", and "APPCPORT".
RALT_UTK_USER_ID	Char	8	467	474	User ID associated with the record.
RALT_UTK_GRP_ID	Char	8	476	483	Group ID associated with the record.
RALT_UTK_DFT_GRP	Yes/No	4	485	488	Is a default group assigned?
RALT_UTK_DFT_SECL	Yes/No	4	490	493	Is a default SECLABEL assigned?
RALT_APPC_LINK	Char	16	495	510	Key to link together APPC records.
RALT_RES_NAME	Char	255	512	766	The resource name.
RALT_SPECIFIED	Char	1024	768	1791	The keywords specified.
RALT_FAILED	Char	1024	1793	2816	The keywords that failed.

Event Qualifiers for RALTER Commands

The event qualifiers that may be associated with a RALTER, ALTDIR, or ALTFILE command are shown in Table 55 on page 181.

Table 55. Event Code Qualifiers for RALTER Command Records

Event Qualifier	Event Number	Event Description
SUCCESS	00	No violations detected.
INSAUTH	01	Insufficient authority.
KEYWVIOL	02	Keyword violation.

The Format of the RDEFINE Record Extension

Table 56 on page 181 describes the format of a record that is created by the RDEFINE, ADDDIR, or ADDFILE command.

Table 56. Format of the RDEFINE Record Extension

Field Name	Type	Length	Position		Comments
			Start	End	
RDEF_CLASS	Char	8	282	289	Class name.
RDEF_OWN_ID	Char	8	291	298	Owner of the profile.
RDEF_USER_NAME	Char	20	300	319	User name.
RDEF_SECL	Char	8	321	328	The SECLABEL associated with the profile.

Table 56. Format of the RDEFINE Record Extension (continued)

Field Name	Type	Length	Position		Comments
			Start	End	
RDEF_UTK_ENCR	Yes/No	4	330	333	Is the UTOKEN associated with this user encrypted?
RDEF_UTK_PRE19	Yes/No	4	335	338	Is this a token for a release earlier than RACF 1.9?
RDEF_UTK_VERPROF	Yes/No	4	340	343	Is the VERIFYX propagation flag set?
RDEF_UTK_NJEUNUSR	Yes/No	4	345	348	Is this the NJE undefined user?
RDEF_UTK_LOGUSR	Yes/No	4	350	353	Is UAUDIT specified for this user?
RDEF_UTK_SPECIAL	Yes/No	4	355	358	Is this a RACF SPECIAL user?
RDEF_UTK_DEFAULT	Yes/No	4	360	363	Is this a default token?
RDEF_UTK_UNKNUSR	Yes/No	4	365	368	Is this an undefined user?
RDEF_UTK_ERROR	Yes/No	4	370	373	Is this user token in error?
RDEF_UTK_TRUSTED	Yes/No	4	375	378	Is this user a part of the trusted computing base (TCB)?
RDEF_UTK_SESSTYPE	Char	8	380	387	The session type of this session.
RDEF_UTK_SURROGAT	Yes/No	4	389	392	Is this a surrogate user?
RDEF_UTK_REMOTE	Yes/No	4	394	397	Is this a remote job?
RDEF_UTK_PRIV	Yes/No	4	399	402	Is this a privileged user ID?
RDEF_UTK_SECL	Char	8	404	411	The SECLABEL of the user.
RDEF_UTK_EXECNODE	Char	8	413	420	The execution node of the work.
RDEF_UTK_SUSER_ID	Char	8	422	429	The submitting user ID.
RDEF_UTK_SNODE	Char	8	431	438	The submitting node.
RDEF_UTK_SGRP_ID	Char	8	440	447	The submitting group ID.
RDEF_UTK_SPOE	Char	8	449	456	The port of entry.
RDEF_UTK_SPCLASS	Char	8	458	465	Class of the POE. Valid values are "TERMINAL", "CONSOLE", "JESINPUT", and "APPCPORT".
RDEF_UTK_USER_ID	Char	8	467	474	User ID associated with the record.
RDEF_UTK_GRP_ID	Char	8	476	483	Group ID associated with the record.
RDEF_UTK_DFT_GRP	Yes/No	4	485	488	Is a default group assigned?
RDEF_UTK_DFT_SECL	Yes/No	4	490	493	Is a default SECLABEL assigned?
RDEF_APPC_LINK	Char	16	495	510	Key to link together APPC records.
RDEF_RES_NAME	Char	255	512	766	The resource name.
RDEF_SPECIFIED	Char	1024	768	1791	The keywords specified.
RDEF_FAILED	Char	1024	1793	2816	The keywords that failed.

Event Qualifiers for RDEFINE Commands

The event qualifiers that may be associated with a RDEFINE, ADDDIR, or ADDFILE command are shown in Table 57 on page 182.

Table 57. Event Code Qualifiers for RDEFINE Command Records

Event Qualifier	Event Number	Event Description
SUCCESS	00	No violations detected.
INSAUTH	01	Insufficient authority.

Table 57. Event Code Qualifiers for RDEFINE Command Records (continued)		
Event Qualifier	**Event Number**	**Event Description**
KEYWVIOL	02	Keyword violation.

The Format of the RDELETE Record Extension

Table 58 on page 183 describes the format of a record that is created by the RDELETE, DELDIR, or DELFILE command.

Table 58. Format of the RDELETE Record Extension

Field Name	Type	Length	Position		Comments
			Start	End	
RDEL_CLASS	Char	8	282	289	Class name.
RDEL_OWN_ID	Char	8	291	298	Owner of the profile.
RDEL_USER_NAME	Char	20	300	319	User name.
RDEL_SECL	Char	8	321	328	The SECLABEL associated with the profile.
RDEL_UTK_ENCR	Yes/No	4	330	333	Is the UTOKEN associated with this user encrypted?
RDEL_UTK_PRE19	Yes/No	4	335	338	Is this a token for a release earlier than RACF 1.9?
RDEL_UTK_VERPROF	Yes/No	4	340	343	Is the VERIFYX propagation flag set?
RDEL_UTK_NJEUNUSR	Yes/No	4	345	348	Is this the NJE undefined user?
RDEL_UTK_LOGUSR	Yes/No	4	350	353	Is UAUDIT specified for this user?
RDEL_UTK_SPECIAL	Yes/No	4	355	358	Is this a RACF SPECIAL user?
RDEL_UTK_DEFAULT	Yes/No	4	360	363	Is this a default token?
RDEL_UTK_UNKNUSR	Yes/No	4	365	368	Is this an undefined user?
RDEL_UTK_ERROR	Yes/No	4	370	373	Is this user token in error?
RDEL_UTK_TRUSTED	Yes/No	4	375	378	Is this user a part of the trusted computing base (TCB)?
RDEL_UTK_SESSTYPE	Char	8	380	387	The session type of this session.
RDEL_UTK_SURROGAT	Yes/No	4	389	392	Is this a surrogate user?
RDEL_UTK_REMOTE	Yes/No	4	394	397	Is this a remote job?
RDEL_UTK_PRIV	Yes/No	4	399	402	Is this a privileged user ID?
RDEL_UTK_SECL	Char	8	404	411	The SECLABEL of the user.
RDEL_UTK_EXECNODE	Char	8	413	420	The execution node of the work.
RDEL_UTK_SUSER_ID	Char	8	422	429	The submitting user ID.
RDEL_UTK_SNODE	Char	8	431	438	The submitting node.
RDEL_UTK_SGRP_ID	Char	8	440	447	The submitting group ID.
RDEL_UTK_SPOE	Char	8	449	456	The port of entry.
RDEL_UTK_SPCLASS	Char	8	458	465	Class of the POE. Valid values are "TERMINAL", "CONSOLE", "JESINPUT", and "APPCPORT".
RDEL_UTK_USER_ID	Char	8	467	474	User ID associated with the record.
RDEL_UTK_GRP_ID	Char	8	476	483	Group ID associated with the record.
RDEL_UTK_DFT_GRP	Yes/No	4	485	488	Is a default group assigned?
RDEL_UTK_DFT_SECL	Yes/No	4	490	493	Is a default SECLABEL assigned?
RDEL_APPC_LINK	Char	16	495	510	Key to link together APPC records.

Table 58. Format of the RDELETE Record Extension (continued)

Field Name	Type	Length	Position		Comments
			Start	End	
RDEL_RES_NAME	Char	255	512	766	The resource name.
RDEL_SPECIFIED	Char	1024	768	1791	The keywords specified.

Event Qualifiers for RDELETE Commands

The event qualifiers that may be associated with a RDELETE, DELDIR, or DELFILE command are shown in Table 59 on page 184.

Table 59. Event Code Qualifiers for RDELETE Command Records

Event Qualifier	Event Number	Event Description
SUCCESS	00	No violations detected.
INSAUTH	01	Insufficient authority.
KEYWVIOL	02	Keyword violation.

The Format of the REMOVE Record Extension

Table 60 on page 184 describes the format of a record that is created by the REMOVE command.

Table 60. Format of the REMOVE Record Extension

Field Name	Type	Length	Position		Comments
			Start	End	
REM_OWN_ID	Char	8	282	289	Owner of the profile.
REM_USER_NAME	Char	20	291	310	User name.
REM_UTK_ENCR	Yes/No	4	312	315	Is the UTOKEN associated with this user encrypted?
REM_UTK_PRE19	Yes/No	4	317	320	Is this a token for a release earlier than RACF 1.9?
REM_UTK_VERPROF	Yes/No	4	322	325	Is the VERIFYX propagation flag set?
REM_UTK_NJEUNUSR	Yes/No	4	327	330	Is this the NJE undefined user?
REM_UTK_LOGUSR	Yes/No	4	332	335	Is UAUDIT specified for this user?
REM_UTK_SPECIAL	Yes/No	4	337	340	Is this a RACF SPECIAL user?
REM_UTK_DEFAULT	Yes/No	4	342	345	Is this a default token?
REM_UTK_UNKNUSR	Yes/No	4	347	350	Is this an undefined user?
REM_UTK_ERROR	Yes/No	4	352	355	Is this user token in error?
REM_UTK_TRUSTED	Yes/No	4	357	360	Is this user a part of the trusted computing base (TCB)?
REM_UTK_SESSTYPE	Char	8	362	369	The session type of this session.
REM_UTK_SURROGAT	Yes/No	4	371	374	Is this a surrogate user?
REM_UTK_REMOTE	Yes/No	4	376	379	Is this a remote job?
REM_UTK_PRIV	Yes/No	4	381	384	Is this a privileged user ID?
REM_UTK_SECL	Char	8	386	393	The SECLABEL of the user.
REM_UTK_EXECNODE	Char	8	395	402	The execution node of the work.
REM_UTK_SUSER_ID	Char	8	404	411	The submitting user ID.
REM_UTK_SNODE	Char	8	413	420	The submitting node.

Table 60. Format of the REMOVE Record Extension (continued)

Field Name	Type	Length	Position		Comments
			Start	End	
REM_UTK_SGRP_ID	Char	8	422	429	The submitting group ID.
REM_UTK_SPOE	Char	8	431	438	The port of entry.
REM_UTK_SPCLASS	Char	8	440	447	Class of the POE. Valid values are "TERMINAL", "CONSOLE", "JESINPUT", and "APPCPORT".
REM_UTK_USER_ID	Char	8	449	456	User ID associated with the record.
REM_UTK_GRP_ID	Char	8	458	465	Group ID associated with the record.
REM_UTK_DFT_GRP	Yes/No	4	467	470	Is a default group assigned?
REM_UTK_DFT_SECL	Yes/No	4	472	475	Is a default SECLABEL assigned?
REM_APPC_LINK	Char	16	477	492	Key to link together APPC records.
REM_USER_ID	Char	8	494	501	The user ID.
REM_SPECIFIED	Char	1024	503	1526	The keywords specified.
REM_FAILED	Char	1024	1528	2551	The keywords that failed.

Event Qualifiers for REMOVE Commands

The event qualifiers that may be associated with a REMOVE command are shown in Table 61 on page 185.

Table 61. Event Code Qualifiers for REMOVE Command Records

Event Qualifier	Event Number	Event Description
SUCCESS	00	No violations detected.
INSAUTH	01	Insufficient authority.
KEYWVIOL	02	Keyword violation.

The Format of the SETROPTS Record Extension

Table 62 on page 185 describes the format of a record that is created by the SETROPTS command.

Table 62. Format of the SETROPTS Record Extension

Field Name	Type	Length	Position		Comments
			Start	End	
SETR_USER_NAME	Char	20	282	301	User name.
SETR_UTK_ENCR	Yes/No	4	303	306	Is the UTOKEN associated with this user encrypted?
SETR_UTK_PRE19	Yes/No	4	308	311	Is this a token for a release earlier than RACF 1.9?
SETR_UTK_VERPROF	Yes/No	4	313	316	Is the VERIFYX propagation flag set?
SETR_UTK_NJEUNUSR	Yes/No	4	318	321	Is this the NJE undefined user?
SETR_UTK_LOGUSR	Yes/No	4	323	326	Is UAUDIT specified for this user?
SETR_UTK_SPECIAL	Yes/No	4	328	331	Is this a RACF SPECIAL user?
SETR_UTK_DEFAULT	Yes/No	4	333	336	Is this a default token?
SETR_UTK_UNKNUSR	Yes/No	4	338	341	Is this an undefined user?
SETR_UTK_ERROR	Yes/No	4	343	346	Is this user token in error?
SETR_UTK_TRUSTED	Yes/No	4	348	351	Is this user a part of the trusted computing base (TCB)?

Table 62. Format of the SETROPTS Record Extension (continued)

Field Name	Type	Length	Position		Comments
			Start	End	
SETR_UTK_SESSTYPE	Char	8	353	360	The session type of this session.
SETR_UTK_SURROGAT	Yes/No	4	362	365	Is this a surrogate user?
SETR_UTK_REMOVE	Yes/No	4	367	370	Is this a remote job?
SETR_UTK_PRIV	Yes/No	4	372	375	Is this a privileged user ID?
SETR_UTK_SECL	Char	8	377	384	The SECLABEL of the user.
SETR_UTK_EXECNODE	Char	8	386	393	The execution node of the work.
SETR_UTK_SUSER_ID	Char	8	395	402	The submitting user ID.
SETR_UTK_SNODE	Char	8	404	411	The submitting node.
SETR_UTK_SGRP_ID	Char	8	413	420	The submitting group ID.
SETR_UTK_SPOE	Char	8	422	429	The port of entry.
SETR_UTK_SPCLASS	Char	8	431	438	Class of the POE. Valid values are "TERMINAL", "CONSOLE", "JESINPUT", and "APPCPORT".
SETR_UTK_USER_ID	Char	8	440	447	User ID associated with the record.
SETR_UTK_GRP_ID	Char	8	449	456	Group ID associated with the record.
SETR_UTK_DFT_GRP	Yes/No	4	458	461	Is a default group assigned?
SETR_UTK_DFT_SECL	Yes/No	4	463	466	Is a default SECLABEL assigned?
SETR_APPC_LINK	Char	16	468	483	Key to link together APPC records.
SETR_SPECIFIED	Char	1024	485	1508	The RRSF keywords specified.

Event Qualifiers for SETROPTS Commands

The event qualifiers that may be associated with a SETROPTS command are shown in Table 63 on page 186.

Table 63. Event Code Qualifiers for SETROPTS Command Records

Event Qualifier	Event Number	Event Description
SUCCESS	00	No violations detected.
INSAUTH	01	Insufficient authority.
KEYWVIOL	02	Keyword violation.

The Format of the RVARY Record Extension

Table 64 on page 186 describes the format of a record that is created by the RVARY command.

Table 64. Format of the RVARY Record Extension

Field Name	Type	Length	Position		Comments
			Start	End	
RVAR_USER_NAME	Char	20	282	301	User name.
RVAR_UTK_ENCR	Yes/No	4	303	306	Is the UTOKEN associated with this user encrypted?
RVAR_UTK_PRE19	Yes/No	4	308	311	Is this a token for a release earlier than RACF 1.9?
RVAR_UTK_VERPROF	Yes/No	4	313	316	Is the VERIFYX propagation flag set?
RVAR_UTK_NJEUNUSR	Yes/No	4	318	321	Is this the NJE undefined user?

Table 64. Format of the RVARY Record Extension (continued)

Field Name	Type	Length	Position		Comments
			Start	End	
RVAR_UTK_LOGUSR	Yes/No	4	323	326	Is UAUDIT specified for this user?
RVAR_UTK_SPECIAL	Yes/No	4	328	331	Is this a RACF SPECIAL user?
RVAR_UTK_DEFAULT	Yes/No	4	333	336	Is this a default token?
RVAR_UTK_UNKNUSR	Yes/No	4	338	341	Is this an undefined user?
RVAR_UTK_ERROR	Yes/No	4	343	346	Is this user token in error?
RVAR_UTK_TRUSTED	Yes/No	4	348	351	Is this user a part of the trusted computing base (TCB)?
RVAR_UTK_SESSTYPE	Char	8	353	360	The session type of this session.
RVAR_UTK_SURROGAT	Yes/No	4	362	365	Is this a surrogate user?
RVAR_UTK_REMOVE	Yes/No	4	367	370	Is this a remote job?
RVAR_UTK_PRIV	Yes/No	4	372	375	Is this a privileged user ID?
RVAR_UTK_SECL	Char	8	377	384	The SECLABEL of the user.
RVAR_UTK_EXECNODE	Char	8	386	393	The execution node of the work.
RVAR_UTK_SUSER_ID	Char	8	395	402	The submitting user ID.
RVAR_UTK_SNODE	Char	8	404	411	The submitting node.
RVAR_UTK_SGRP_ID	Char	8	413	420	The submitting group ID.
RVAR_UTK_SPOE	Char	8	422	429	The port of entry.
RVAR_UTK_SPCLASS	Char	8	431	438	Class of the POE. Valid values are "TERMINAL", "CONSOLE", "JESINPUT", and "APPCPORT".
RVAR_UTK_USER_ID	Char	8	440	447	User ID associated with the record.
RVAR_UTK_GRP_ID	Char	8	449	456	Group ID associated with the record.
RVAR_UTK_DFT_GRP	Yes/No	4	458	461	Is a default group assigned?
RVAR_UTK_DFT_SECL	Yes/No	4	463	466	Is a default SECLABEL assigned?
RVAR_APPC_LINK	Char	16	468	483	Key to link together APPC records.

Event Qualifiers for RVARY Commands

The event qualifiers that may be associated with a RVARY command are shown in Table 65 on page 187.

Table 65. Event Code Qualifiers for RVARY Command Records

Event Qualifier	Event Number	Event Description
SUCCESS	00	No violations detected.
INSAUTH	01	Insufficient authority.
KEYWVIOL	02	Keyword violation.

The Format of the APPCLU Record Extension

Table 66 on page 187 describes the format of a record that is created by the define resource operation.

Table 66. Format of the APPCLU Record Extension

Field Name	Type	Length	Position		Comments
			Start	End	
APPC_RES_NAME	Char	255	282	536	Resource name.

Table 66. Format of the APPCLU Record Extension (continued)

Field Name	Type	Length	Position		Comments
			Start	End	
APPC_CLASS	Char	8	538	545	Class name.
APPC_TYPE	Char	8	547	554	Type of resource data. Valid values are "RESOURCE" if ACC_NAME is a generic resource name, and "PROFILE" if ACC_NAME is a generic profile.
APPC_NAME	Char	246	556	801	Resource or profile name.
APPC_OWN_ID	Char	8	803	810	Name of the profile owner.
APPC_USER_NAME	Char	20	812	831	User name.
APPC_UTK_ENCR	Yes/No	4	833	836	Is the UTOKEN associated with this user encrypted?
APPC_UTK_PRE19	Yes/No	4	838	841	Is this a token for a release earlier than RACF 1.9?
APPC_UTK_VERPROF	Yes/No	4	843	846	Is the VERIFYX propagation flag set?
APPC_UTK_NJEUNUSR	Yes/No	4	848	851	Is this the NJE undefined user?
APPC_UTK_LOGUSR	Yes/No	4	853	856	Is UAUDIT specified for this user?
APPC_UTK_SPECIAL	Yes/No	4	858	861	Is this a RACF SPECIAL user?
APPC_UTK_DEFAULT	Yes/No	4	863	866	Is this a default token?
APPC_UTK_UNKNUSR	Yes/No	4	868	871	Is this an undefined user?
APPC_UTK_ERROR	Yes/No	4	873	876	Is this user token in error?
APPC_UTK_TRUSTED	Yes/No	4	878	881	Is this user a part of the trusted computing base (TCB)?
APPC_UTK_SESSTYPE	Char	8	883	890	The session type of this session.
APPC_UTK_SURROGAT	Yes/No	4	892	895	Is this a surrogate user?
APPC_UTK_REMOTE	Yes/No	4	897	900	Is this a remote job?
APPC_UTK_PRIV	Yes/No	4	902	905	Is this a privileged user ID?
APPC_UTK_SECL	Char	8	907	914	The SECLABEL of the user.
APPC_UTK_EXECNODE	Char	8	916	923	The execution node of the work.
APPC_UTK_SUSER_ID	Char	8	925	932	The submitting user ID.
APPC_UTK_SNODE	Char	8	934	941	The submitting node.
APPC_UTK_SGRP_ID	Char	8	943	950	The submitting group ID.
APPC_UTK_SPOE	Char	8	952	959	The port of entry.
APPC_UTK_SPCLASS	Char	8	961	968	Class of the POE. Valid values are "TERMINAL", "CONSOLE", "JESINPUT", and "APPCPORT".
APPC_UTK_USER_ID	Char	8	970	977	User ID associated with the record.
APPC_UTK_GRP_ID	Char	8	979	986	Group ID associated with the record.
APPC_UTK_DFT_GRP	Yes/No	4	988	991	Is a default group assigned?
APPC_UTK_DFT_SECL	Yes/No	4	993	996	Is a default SECLABEL assigned?
APPC_APPC_LINK	Char	16	998	1013	Key to link together APPC records.

Event Qualifiers for APPCLU Requests

The event qualifiers that may be associated with a APPCLU event are shown in Table 67 on page 189.

Table 67. Event Code Qualifiers for APPCLU Records

Event Qualifier	Event Number	Event Description
SUCCESS	00	Partner verification OK.
NOVERIFY	01	Session established without verification.
LKEYEXPR	02	Local key expires in less than 5 days.
REVOKED	03	Partner LU access has been revoked.
NOMATCH	04	Partner LU key does not match this LU key.
TRMSECUR	05	Session terminated for security reasons.
NOSESKEY	06	Required session key not defined.
LUATTACK	07	Possible security attack by partner LU.
NOPRTKEY	08	Session key not defined for the partner LU.
NOKEY	09	Session key not defined for this LU.
SNAERROR	10	SNA security-related session error.
PROFCHNG	11	Profile changed during verification.
SKEYEXPR	12	Expired session key.

The Format of the General Event Record Extension

Table 68 on page 189 describes the format of a record that is created by a general event.

Table 68. Format of the General Event Record Extension

Field Name	Type	Length	Position		Comments
			Start	End	
GEN_CLASS	Char	8	282	289	Class name.
GEN_LOGSTR	Char	255	291	545	LOGSTR= data from the RACROUTE
GEN_USER_NAME	Char	20	547	566	User name.
GEN_UTK_ENCR	Yes/No	4	568	571	Is the UTOKEN associated with this user encrypted?
GEN_UTK_PRE19	Yes/No	4	573	576	Is this a token for a release earlier than RACF 1.9?
GEN_UTK_VERPROF	Yes/No	4	578	581	Is the VERIFYX propagation flag set?
GEN_UTK_NJEUNUSR	Yes/No	4	583	586	Is this the NJE undefined user?
GEN_UTK_LOGUSR	Yes/No	4	588	591	Is UAUDIT specified for this user?
GEN_UTK_SPECIAL	Yes/No	4	593	596	Is this a RACF SPECIAL user?
GEN_UTK_DEFAULT	Yes/No	4	598	601	Is this a default token?
GEN_UTK_UNKNUSR	Yes/No	4	603	606	Is this an undefined user?
GEN_UTK_ERROR	Yes/No	4	608	611	Is this user token in error?
GEN_UTK_TRUSTED	Yes/No	4	613	616	Is this user a part of the trusted computing base (TCB)?
GEN_UTK_SESSTYPE	Char	8	618	625	The session type of this session.
GEN_UTK_SURROGAT	Yes/No	4	627	630	Is this a surrogate user?
GEN_UTK_REMOTE	Yes/No	4	632	635	Is this a remote job?
GEN_UTK_PRIV	Yes/No	4	637	640	Is this a privileged user ID?
GEN_UTK_SECL	Char	8	642	649	The SECLABEL of the user.
GEN_UTK_EXECNODE	Char	8	651	658	The execution node of the work.

Table 68. Format of the General Event Record Extension (continued)

Field Name	Type	Length	Position		Comments
			Start	End	
GEN_UTK_SUSER_ID	Char	8	660	667	The submitting user ID.
GEN_UTK_SNODE	Char	8	669	676	The submitting node.
GEN_UTK_SGRP_ID	Char	8	678	685	The submitting group ID.
GEN_UTK_SPOE	Char	8	687	694	The port of entry.
GEN_UTK_SPCLASS	Char	8	696	703	Class of the POE. Valid values are "TERMINAL", "CONSOLE", "JESINPUT", and "APPCPORT".
GEN_UTK_USER_ID	Char	8	705	712	User ID associated with the record.
GEN_UTK_GRP_ID	Char	8	714	721	Group ID associated with the record.
GEN_UTK_DFT_GRP	Yes/No	4	723	726	Is a default group assigned?
GEN_UTK_DFT_SECL	Yes/No	4	728	731	Is a default SECLABEL assigned?
GEN_APPC_LINK	Char	16	733	748	Key to link together GENERAL records.

Event Qualifiers for General Events

The event qualifiers that may be associated with a General Event are determined by the installation. These event codes will be unloaded as integer values.

The Format of the Directory Search Record Extension

Table 69 on page 190 describes the format of a record that is created by a directory search event.

Table 69. Format of the Directory Search Record Extension

Field Name	Type	Length	Position		Comments
			Start	End	
DSCH_CLASS	Char	8	282	289	Class name.
DSCH_USER_NAME	Char	20	291	310	The name associated with the user ID.
DSCH_UTK_ENCR	Yes/No	4	312	315	Is the UTOKEN associated with this user encrypted?
DSCH_UTK_PRE19	Yes/No	4	317	320	Is this a token for a release earlier than RACF 1.9?
DSCH_UTK_VERPROF	Yes/No	4	322	325	Is the VERIFYX propagation flag set?
DSCH_UTK_NJEUNUSR	Yes/No	4	327	330	Is this the NJE undefined user?
DSCH_UTK_LOGUSR	Yes/No	4	332	335	Is UAUDIT specified for this user?
DSCH_UTK_SPECIAL	Yes/No	4	337	340	Is this a RACF SPECIAL user?
DSCH_UTK_DEFAULT	Yes/No	4	342	345	Is this a default token?
DSCH_UTK_UNKNUSR	Yes/No	4	347	350	Is this an undefined user?
DSCH_UTK_ERROR	Yes/No	4	352	355	Is this user token in error?
DSCH_UTK_TRUSTED	Yes/No	4	357	360	Is this user a part of the trusted computing base (TCB)?
DSCH_UTK_SESSTYPE	Char	8	362	369	The session type of this session.
DSCH_UTK_SURROGAT	Yes/No	4	371	374	Is this a surrogate user?
DSCH_UTK_REMOTE	Yes/No	4	376	379	Is this a remote job?
DSCH_UTK_PRIV	Yes/No	4	381	384	Is this a privileged user ID?
DSCH_UTK_SECL	Char	8	386	393	The SECLABEL of the user.
DSCH_UTK_EXECNODE	Char	8	395	402	The execution node of the work.

Table 69. Format of the Directory Search Record Extension (continued)

Field Name	Type	Length	Position		Comments
			Start	End	
DSCH_UTK_SUSER_ID	Char	8	404	411	The submitting user ID.
DSCH_UTK_SNODE	Char	8	413	420	The submitting node.
DSCH_UTK_SGRP_ID	Char	8	422	429	The submitting group ID.
DSCH_UTK_SPOE	Char	8	431	438	The port of entry.
DSCH_UTK_SPCLASS	Char	8	440	447	Class of the POE. Valid values are "TERMINAL", "CONSOLE", "JESINPUT", and "APPCPORT".
DSCH_UTK_USER_ID	Char	8	449	456	User ID associated with the record.
DSCH_UTK_GRP_ID	Char	8	458	465	Group ID associated with the record.
DSCH_UTK_DFT_GRP	Yes/No	4	467	470	Is a default group assigned?
DSCH_UTK_DFT_SECL	Yes/No	4	472	475	Is a default SECLABEL assigned?
DSCH_APPC_LINK	Char	16	477	492	A key to link together audit records for a user's APPC transaction processing work.
DSCH_AUDIT_CODE	Char	11	494	504	Audit function code. For more information on the function codes, see Appendix I, “OpenExtensions Audit Function Codes,” on page 439.
DSCH_OLD_REAL_UID	Integer	10	506	515	Old real UID.
DSCH_OLD_EFF_UID	Integer	10	517	526	Old effective UID.
DSCH_OLD_SAVED_UID	Integer	10	528	537	Old saved UID.
DSCH_OLD_REAL_GID	Integer	10	539	548	Old real GID.
DSCH_OLD_EFF_GID	Integer	10	550	559	Old effective GID.
DSCH_OLD_SAVED_GID	Integer	10	561	570	Old saved GID.
DSCH_PATH_NAME	Char	1023	572	1594	The requested path name.
DSCH_FILE_ID	Char	32	1596	1627	File ID.
DSCH_FILE_OWN_UID	Integer	10	1629	1638	The owner UID associated with the file.
DSCH_FILE_OWN_GID	Integer	10	1640	1649	The owner GID associated with the file.
DSCH_REQUEST_READ	Yes/No	4	1651	1654	Did the requested access include read?
DSCH_REQUEST_WRITE	Yes/No	4	1656	1659	Did the requested access include write?
DSCH_REQUEST_EXEC	Yes/No	4	1661	1664	Did the requested access include EXECUTE?
DSCH_REQUEST_DSRCH	Yes/No	4	1666	1669	Did the requested access include directory search?
DSCH_ACCESS_TYPE	Char	8	1671	1678	What bits were used in granting the access? Valid values are "OWNER", "GROUP", "NO", and "OTHER".
DSCH_ALLOWED_READ	Yes/No	4	1680	1683	Was read access allowed?
DSCH_ALLOWED_WRITE	Yes/No	4	1685	1688	Was write access allowed?
DSCH_ALLOWED_EXEC	Yes/No	4	1690	1693	Was execute or search access allowed?
DSCH_REQUEST_PATH2	Char	1023	1695	2717	Second requested path name.
DSCH_SERVICE_CODE	Char	11	2719	2729	The service that was being processed. This is set only when the DSCH_AUDIT_CODE is "LOOKUP". For more information on the function codes, see Appendix I, “OpenExtensions Audit Function Codes,” on page 439.
DSCH_HFS_DS_NAME	Char	44	2731	2774	HFS data set name for the mounted file system.
DSCH_SYMLINK	Char	1023	2776	3798	The content of SYMLINK.
DSCH_FILE_NAME	Char	256	3800	4055	The file name that is being checked.

Table 69. Format of the Directory Search Record Extension (continued)

Field Name	Type	Length	Position		Comments
			Start	End	
DSCH_PATH_TYPE	Char	4	4057	4060	Type of the requested path name. Valid values are "OLD" and "NEW".
DSCH_FILEPOOL	Char	8	4062	4069	SFS filepool containing the BFS file.
DSCH_FILESPACE	Char	8	4071	4078	SFS filespace containing the BFS file.
DSCH_INODE	Integer	10	4080	4089	Inode (file serial number).
DSCH_SCID	Integer	10	4091	4100	File SCID.
DSCH_DCE_LINK	Char	16	4102	4117	Link to connect DCE records that originate from a single DCE request.
DSCH_AUTH_TYPE	Char	13	4119	4131	Defines the type of request. Valid values are: "SERVER" and "AUTH_CLIENT" and "UNAUTH_CLIENT".

Event Qualifiers for Directory Search Requests

The event qualifiers that may be associated with a directory search event are shown in Table 70 on page 192.

Table 70. Event Code Qualifiers for Directory Search Records

Event Qualifier	Event Number	Event Description
SUCCESS	00	Access allowed.
NOTAUTH	01	Not authorized to search the directory.

The Format of the Check Directory Access Record Extension

Table 71 on page 192 describes the format of a record that is created by checking access to a directory.

Table 71. Format of the Check Directory Access Record Extension

Field Name	Type	Length	Position		Comments
			Start	End	
DACC_CLASS	Char	8	282	289	Class name.
DACC_USER_NAME	Char	20	291	310	The name associated with the user ID.
DACC_UTK_ENCR	Yes/No	4	312	315	Is the UTOKEN associated with this user encrypted?
DACC_UTK_PRE19	Yes/No	4	317	320	Is this a token for a release earlier than RACF 1.9?
DACC_UTK_VERPROF	Yes/No	4	322	325	Is the VERIFYX propagation flag set?
DACC_UTK_NJEUNUSR	Yes/No	4	327	330	Is this the NJE undefined user?
DACC_UTK_LOGUSR	Yes/No	4	332	335	Is UAUDIT specified for this user?
DACC_UTK_SPECIAL	Yes/No	4	337	340	Is this a RACF SPECIAL user?
DACC_UTK_DEFAULT	Yes/No	4	342	345	Is this a default token?
DACC_UTK_UNKNUSR	Yes/No	4	347	350	Is this an undefined user?
DACC_UTK_ERROR	Yes/No	4	352	355	Is this user token in error?
DACC_UTK_TRUSTED	Yes/No	4	357	360	Is this user a part of the trusted computing base (TCB)?
DACC_UTK_SESSTYPE	Char	8	362	369	The session type of this session.
DACC_UTK_SURROGAT	Yes/No	4	371	374	Is this a surrogate user?

Table 71. Format of the Check Directory Access Record Extension (continued)

Field Name	Type	Length	Position		Comments
			Start	End	
DACC_UTK_REMOTE	Yes/No	4	376	379	Is this a remote job?
DACC_UTK_PRIV	Yes/No	4	381	384	Is this a privileged user ID?
DACC_UTK_SECL	Char	8	386	393	The SECLABEL of the user.
DACC_UTK_EXECNODE	Char	8	395	402	The execution node of the work.
DACC_UTK_SUSER_ID	Char	8	404	411	The submitting user ID.
DACC_UTK_SNODE	Char	8	413	420	The submitting node.
DACC_UTK_SGRP_ID	Char	8	422	429	The submitting group ID.
DACC_UTK_SPOE	Char	8	431	438	The port of entry.
DACC_UTK_SPCLASS	Char	8	440	447	Class of the POE. Valid values are "TERMINAL", "CONSOLE", "JESINPUT", and "APPCPORT".
DACC_UTK_USER_ID	Char	8	449	456	User ID associated with the record.
DACC_UTK_GRP_ID	Char	8	458	465	Group ID associated with the record.
DACC_UTK_DFT_GRP	Yes/No	4	467	470	Is a default group assigned?
DACC_UTK_DFT_SECL	Yes/No	4	472	475	Is a default SECLABEL assigned?
DACC_APPC_LINK	Char	16	477	492	Key to link together APPC records.
DACC_AUDIT_CODE	Char	11	494	504	Audit function code. For more information on the function codes, see Appendix I, "OpenExtensions Audit Function Codes," on page 439.
DACC_OLD_REAL_UID	Integer	10	506	515	Old real UID.
DACC_OLD_EFF_UID	Integer	10	517	526	Old effective UID.
DACC_OLD_SAVED_UID	Integer	10	528	537	Old saved UID.
DACC_OLD_REAL_GID	Integer	10	539	548	Old real GID.
DACC_OLD_EFF_GID	Integer	10	550	559	Old effective GID.
DACC_OLD_SAVED_GID	Integer	10	561	570	Old saved GID.
DACC_PATH_NAME	Char	1023	572	1594	The requested path name.
DACC_FILE_ID	Char	32	1596	1627	File ID.
DACC_FILE_OWN_UID	Integer	10	1629	1638	The owner UID associated with the file.
DACC_FILE_OWN_GID	Integer	10	1640	1649	The owner GID associated with the file.
DACC_REQUEST_READ	Yes/No	4	1651	1654	Did the requested access include read?
DACC_REQUEST_WRITE	Yes/No	4	1656	1659	Did the requested access include write?
DACC_REQUEST_EXEC	Yes/No	4	1661	1664	Did the requested access include execute?
DACC_REQUEST_DSRCH	Yes/No	4	1666	1669	Did the requested access include directory search?
DACC_ACCESS_TYPE	Char	8	1671	1678	What bits were used in granting the access? Valid values are "OWNER", "GROUP", "NO", and "OTHER".
DACC_ALLOWED_READ	Yes/No	4	1680	1683	Was read access allowed?
DACC_ALLOWED_WRITE	Yes/No	4	1685	1688	Was write access allowed?
DACC_ALLOWED_EXEC	Yes/No	4	1690	1693	Was execute access allowed?
DACC_REQUEST_PATH2	Char	1023	1695	2717	Second requested path name.
DACC_SYMLINK	Char	1023	2719	3741	The content of SYMLINK.
DACC_FILE_NAME	Char	256	3743	3998	The file name that is being checked.

Table 71. Format of the Check Directory Access Record Extension (continued)

Field Name	Type	Length	Position		Comments
			Start	End	
DACC_PATH_TYPE	Char	4	4000	4003	Type of the requested path name. Valid values are "OLD" and "NEW".
DACC_FILEPOOL	Char	8	4005	4012	SFS filepool containing the BFS file.
DACC_FILESPACE	Char	8	4014	4021	SFS filespace containing the BFS file.
DACC_INODE	Integer	10	4023	4032	Inode (file serial number).
DACC_SCID	Integer	10	4034	4043	File SCID.
DACC_DCE_LINK	Char	16	4045	4060	Link to connect DCE records that originate from a single DCE request.
DACC_AUTH_TYPE	Char	13	4062	4074	Defines the type of request. Valid values are: "SERVER" and "AUTH_CLIENT" and "UNAUTH_CLIENT".

Event Qualifiers for Check Directory Access Requests

The event qualifiers that may be associated with a directory search event are shown in Table 72 on page 194.

Table 72. Event Code Qualifiers for Check Directory Access Records

Event Qualifier	Event Number	Event Description
SUCCESS	00	Access allowed.
NOTAUTH	01	Not authorized to the directory.

The Format of the Check File Access Record Extension

Table 73 on page 194 describes the format of a record that is created by checking access to a file.

Table 73. Format of the Check File Access Record Extension

Field Name	Type	Length	Position		Comments
			Start	End	
FACC_CLASS	Char	8	282	289	Class name.
FACC_USER_NAME	Char	20	291	310	The name associated with the user ID.
FACC_UTK_ENCR	Yes/No	4	312	315	Is the UTOKEN associated with this user encrypted?
FACC_UTK_PRE19	Yes/No	4	317	320	Is this a token for a release earlier than RACF 1.9?
FACC_UTK_VERPROF	Yes/No	4	322	325	Is the VERIFYX propagation flag set?
FACC_UTK_NJEUNUSR	Yes/No	4	327	330	Is this the NJE undefined user?
FACC_UTK_LOGUSR	Yes/No	4	332	335	Is UAUDIT specified for this user?
FACC_UTK_SPECIAL	Yes/No	4	337	340	Is this a RACF SPECIAL user?
FACC_UTK_DEFAULT	Yes/No	4	342	345	Is this a default token?
FACC_UTK_UNKNUSR	Yes/No	4	347	350	Is this an undefined user?
FACC_UTK_ERROR	Yes/No	4	352	355	Is this user token in error?
FACC_UTK_TRUSTED	Yes/No	4	357	360	Is this user a part of the trusted computing base (TCB)?
FACC_UTK_SESSTYPE	Char	8	362	369	The session type of this session.
FACC_UTK_SURROGAT	Yes/No	4	371	374	Is this a surrogate user?

Table 73. Format of the Check File Access Record Extension (continued)

Field Name	Type	Length	Position		Comments
			Start	End	
FACC_UTK_REMOTE	Yes/No	4	376	379	Is this a remote job?
FACC_UTK_PRIV	Yes/No	4	381	384	Is this a privileged user ID?
FACC_UTK_SECL	Char	8	386	393	The SECLABEL of the user.
FACC_UTK_EXECNODE	Char	8	395	402	The execution node of the work.
FACC_UTK_SUSER_ID	Char	8	404	411	The submitting user ID.
FACC_UTK_SNODE	Char	8	413	420	The submitting node.
FACC_UTK_SGRP_ID	Char	8	422	429	The submitting group ID.
FACC_UTK_SPOE	Char	8	431	438	The port of entry.
FACC_UTK_SPCLASS	Char	8	440	447	Class of the POE. Valid values are "TERMINAL", "CONSOLE", "JESINPUT", and "APPCPORT".
FACC_UTK_USER_ID	Char	8	449	456	User ID associated with the record.
FACC_UTK_GRP_ID	Char	8	458	465	Group ID associated with the record.
FACC_UTK_DFT_GRP	Yes/No	4	467	470	Is a default group assigned?
FACC_UTK_DFT_SECL	Yes/No	4	472	475	Is a default SECLABEL assigned?
FACC_APPC_LINK	Char	16	477	492	Key to link together APPC records.
FACC_AUDIT_CODE	Char	11	494	504	Audit function code. For more information on the function codes, see Appendix I, "OpenExtensions Audit Function Codes," on page 439.
FACC_OLD_REAL_UID	Integer	10	506	515	Old real UID.
FACC_OLD_EFF_UID	Integer	10	517	526	Old effective UID.
FACC_OLD_SAVED_UID	Integer	10	528	537	Old saved UID.
FACC_OLD_REAL_GID	Integer	10	539	548	Old real GID.
FACC_OLD_EFF_GID	Integer	10	550	559	Old effective GID.
FACC_OLD_SAVED_GID	Integer	10	561	570	Old saved GID.
FACC_PATH_NAME	Char	1023	572	1594	The requested path name.
FACC_FILE_ID	Char	32	1596	1627	File ID.
FACC_FILE_OWN_UID	Integer	10	1629	1638	The owner UID associated with the file.
FACC_FILE_OWN_GID	Integer	10	1640	1649	The owner GID associated with the file.
FACC_REQUEST_READ	Yes/No	4	1651	1654	Did the requested access include read?
FACC_REQUEST_WRITE	Yes/No	4	1656	1659	Did the requested access include write?
FACC_REQUEST_EXEC	Yes/No	4	1661	1664	Did the requested access include EXECUTE?
FACC_REQUEST_DSRCH	Yes/No	4	1666	1669	Did the requested access include directory search?
FACC_ACCESS_TYPE	Char	8	1671	1678	What bits were used in granting the access? Valid values are "OWNER", "GROUP", "NO", and "OTHER".
FACC_ALLOWED_READ	Yes/No	4	1680	1683	Was read access allowed?
FACC_ALLOWED_WRITE	Yes/No	4	1685	1688	Was write access allowed?
FACC_ALLOWED_EXEC	Yes/No	4	1690	1693	Was execute access allowed?
FACC_REQUEST_PATH2	Char	1023	1695	2717	Second requested path name.
FACC_FILE_NAME	Char	256	2719	2974	The file name that is being checked.
FACC_PATH_TYPE	Char	4	2976	2979	Type of the requested path name. Valid values are "OLD" and "NEW".

Table 73. Format of the Check File Access Record Extension (continued)					
Field Name	**Type**	**Length**	**Position**		**Comments**
			Start	**End**	
FACC_FILEPOOL	Char	8	2981	2988	SFS filepool containing the BFS file.
FACC_FILESPACE	Char	8	2990	2997	SFS filespace containing the BFS file.
FACC_INODE	Integer	10	2999	3008	Inode (file serial number).
FACC_SCID	Integer	10	3010	3019	File SCID.
FACC_DCE_LINK	Char	16	3021	3036	Link to connect DCE records that originate from a single DCE request.
FACC_AUTH_TYPE	Char	13	3038	3050	Defines the type of request. Valid values are: "SERVER" and "AUTH_CLIENT" and "UNAUTH_CLIENT".

Event Qualifiers for Check File Access Requests

The event qualifiers that may be associated with a check file access event are shown in Table 74 on page 196.

Table 74. Event Code Qualifiers for Check File Access Records		
Event Qualifier	**Event Number**	**Event Description**
SUCCESS	00	Access allowed.
NOTAUTH	01	Not authorized to the file.

The Format of the Change Audit Record Extension

Table 75 on page 196 describes the format of a record that is created by checking access to a file.

Table 75. Format of the Change Audit Record Extension					
Field Name	**Type**	**Length**	**Position**		**Comments**
			Start	**End**	
CAUD_CLASS	Char	8	282	289	Class name.
CAUD_USER_NAME	Char	20	291	310	The name associated with the user ID.
CAUD_UTK_ENCR	Yes/No	4	312	315	Is the UTOKEN associated with this user encrypted?
CAUD_UTK_PRE19	Yes/No	4	317	320	Is this a token for a release earlier than RACF 1.9?
CAUD_UTK_VERPROF	Yes/No	4	322	325	Is the VERIFYX propagation flag set?
CAUD_UTK_NJEUNUSR	Yes/No	4	327	330	Is this the NJE undefined user?
CAUD_UTK_LOGUSR	Yes/No	4	332	335	Is UAUDIT specified for this user?
CAUD_UTK_SPECIAL	Yes/No	4	337	340	Is this a RACF SPECIAL user?
CAUD_UTK_DEFAULT	Yes/No	4	342	345	Is this a default token?
CAUD_UTK_UNKNUSR	Yes/No	4	347	350	Is this an undefined user?
CAUD_UTK_ERROR	Yes/No	4	352	355	Is this user token in error?
CAUD_UTK_TRUSTED	Yes/No	4	357	360	Is this user a part of the trusted computing base (TCB)?
CAUD_UTK_SESSTYPE	Char	8	362	369	The session type of this session.
CAUD_UTK_SURROGAT	Yes/No	4	371	374	Is this a surrogate user?
CAUD_UTK_REMOTE	Yes/No	4	376	379	Is this a remote job?

Table 75. Format of the Change Audit Record Extension (continued)

Field Name	Type	Length	Position		Comments
			Start	End	
CAUD_UTK_PRIV	Yes/No	4	381	384	Is this a privileged user ID?
CAUD_UTK_SECL	Char	8	386	393	The SECLABEL of the user.
CAUD_UTK_EXECNODE	Char	8	395	402	The execution node of the work.
CAUD_UTK_SUSER_ID	Char	8	404	411	The submitting user ID.
CAUD_UTK_SNODE	Char	8	413	420	The submitting node.
CAUD_UTK_SGRP_ID	Char	8	422	429	The submitting group ID.
CAUD_UTK_SPOE	Char	8	431	438	The port of entry.
CAUD_UTK_SPCLASS	Char	8	440	447	Class of the POE. Valid values are "TERMINAL", "CONSOLE", "JESINPUT", and "APPCPORT".
CAUD_UTK_USER_ID	Char	8	449	456	User ID associated with the record.
CAUD_UTK_GRP_ID	Char	8	458	465	Group ID associated with the record.
CAUD_UTK_DFT_GRP	Yes/No	4	467	470	Is a default group assigned?
CAUD_UTK_DFT_SECL	Yes/No	4	472	475	Is a default SECLABEL assigned?
CAUD_APPC_LINK	Char	16	477	492	A key to link together audit records for a user's APPC transaction processing work.
CAUD_AUDIT_CODE	Char	11	494	504	Audit function code. For more information on the function codes, see Appendix I, "OpenExtensions Audit Function Codes," on page 439.
CAUD_OLD_REAL_UID	Integer	10	506	515	Old real UID.
CAUD_OLD_EFF_UID	Integer	10	517	526	Old effective UID.
CAUD_OLD_SAVED_UID	Integer	10	528	537	Old saved UID.
CAUD_OLD_REAL_GID	Integer	10	539	548	Old real GID.
CAUD_OLD_EFF_GID	Integer	10	550	559	Old effective GID.
CAUD_OLD_SAVED_GID	Integer	10	561	570	Old saved GID.
CAUD_PATH_NAME	Char	1023	572	1594	The requested path name.
CAUD_FILE_ID	Char	32	1596	1627	File ID.
CAUD_FILE_OWN_UID	Integer	10	1629	1638	The owner UID associated with the file.
CAUD_FILE_OWN_GID	Integer	10	1640	1649	The owner GID associated with the file.
CAUD_REQUEST_READ	Char	8	1651	1658	What audit options are requested for a READ operation? Valid values are "NONE", "SUCCESS", "FAIL", and "ALL".
CAUD_REQUEST_WRITE	Char	8	1660	1667	What audit options are requested for a WRITE operation? Valid values are "NONE", "SUCCESS", "FAIL", and "ALL".
CAUD_REQUEST_EXEC	Char	8	1669	1676	What audit options are requested for an EXECUTE operation? Valid values are "NONE", "SUCCESS", "FAIL", and "ALL".
CAUD_UOLD_READ	Char	8	1678	1685	What were the previous user audit options for READ actions? Valid values are "NONE", "SUCCESS", "FAIL", and "ALL".
CAUD_UOLD_WRITE	Char	8	1687	1694	What were the previous user audit options for WRITE actions? Valid values are "NONE", "SUCCESS", "FAIL", and "ALL".

Table 75. Format of the Change Audit Record Extension (continued)

Field Name	Type	Length	Position		Comments
			Start	End	
CAUD_UOLD_EXEC	Char	8	1696	1703	What were the previous user audit options for EXECUTE actions? Valid values are "NONE", "SUCCESS", "FAIL", and "ALL".
CAUD_AOLD_READ	Char	8	1705	1712	What were the previous auditor audit options for READ actions? Valid values are "NONE", "SUCCESS", "FAIL", and "ALL".
CAUD_AOLD_WRITE	Char	8	1714	1721	What were the previous auditor audit options for WRITE actions? Valid values are "NONE", "SUCCESS", "FAIL", and "ALL".
CAUD_AOLD_EXEC	Char	8	1723	1730	What were the previous auditor audit options for EXECUTE actions? Valid values are "NONE", "SUCCESS", "FAIL", and "ALL".
CAUD_UNEW_READ	Char	8	1732	1739	What are the new user audit options for READ actions? Valid values are "NONE", "SUCCESS", "FAIL", and "ALL".
CAUD_UNEW_WRITE	Char	8	1741	1748	What are the new user audit options for WRITE actions? Valid values are "NONE", "SUCCESS", "FAIL", and "ALL".
CAUD_UNEW_EXEC	Char	8	1750	1757	What are the new user audit options for EXECUTE actions? Valid values are "NONE", "SUCCESS", "FAIL", and "ALL".
CAUD_ANEW_READ	Char	8	1759	1766	What are the new auditor audit options for READ actions? Valid values are "NONE", "SUCCESS", "FAIL", and "ALL".
CAUD_ANEW_WRITE	Char	8	1768	1775	What are the new auditor audit options for WRITE actions? Valid values are "NONE", "SUCCESS", "FAIL", and "ALL".
CAUD_ANEW_EXEC	Char	8	1777	1784	What are the new auditor audit options for EXECUTE actions? Valid values are "NONE", "SUCCESS", "FAIL", and "ALL".
CAUD_FILEPOOL	Char	8	1786	1793	SFS filepool containing the BFS file.
CAUD_FILESPACE	Char	8	1795	1802	SFS filespace containing the BFS file.
CAUD_INODE	Integer	10	1804	1813	Inode (file serial number).
CAUD_SCID	Integer	10	1815	1824	File SCID.
CAUD_DCE_LINK	Char	16	1826	1841	Link to connect DCE records that originate from a single DCE request.
CAUD_AUTH_TYPE	Char	13	1843	1855	Defines the type of request. Valid values are: "SERVER" and "AUTH_CLIENT" and "UNAUTH_CLIENT".

Event Qualifiers for Change Audit Requests

The event qualifiers that may be associated with a directory search event are shown in Table 76 on page 198.

Table 76. Event Code Qualifiers for Change Audit Records

Event Qualifier	Event Number	Event Description
SUCCESS	00	File's audit options changed.
NOTAUTHU	01	Not authorized to change the user audit options on the specified file.
NOTAUTHA	02	Not authorized to change the auditor audit options on the specified file.

The Format of the Change Directory Record Extension

Table 77 on page 199 describes the format of a record that is created by changing directories.

Table 77. Format of the Change Directory Record Extension

Field Name	Type	Length	Position		Comments
			Start	End	
CDIR_CLASS	Char	8	282	289	Class name.
CDIR_USER_NAME	Char	20	291	310	The name associated with the user ID.
CDIR_UTK_ENCR	Yes/No	4	312	315	Is the UTOKEN associated with this user encrypted?
CDIR_UTK_PRE19	Yes/No	4	317	320	Is this a token for a release earlier than RACF 1.9?
CDIR_UTK_VERPROF	Yes/No	4	322	325	Is the VERIFYX propagation flag set?
CDIR_UTK_NJEUNUSR	Yes/No	4	327	330	Is this the NJE undefined user?
CDIR_UTK_LOGUSR	Yes/No	4	332	335	Is UAUDIT specified for this user?
CDIR_UTK_SPECIAL	Yes/No	4	337	340	Is this a RACF SPECIAL user?
CDIR_UTK_DEFAULT	Yes/No	4	342	345	Is this a default token?
CDIR_UTK_UNKNUSR	Yes/No	4	347	350	Is this an undefined user?
CDIR_UTK_ERROR	Yes/No	4	352	355	Is this user token in error?
CDIR_UTK_TRUSTED	Yes/No	4	357	360	Is this user a part of the trusted computing base (TCB)?
CDIR_UTK_SESSTYPE	Char	8	362	369	The session type of this session.
CDIR_UTK_SURROGAT	Yes/No	4	371	374	Is this a surrogate user?
CDIR_UTK_REMOTE	Yes/No	4	376	379	Is this a remote job?
CDIR_UTK_PRIV	Yes/No	4	381	384	Is this a privileged user ID?
CDIR_UTK_SECL	Char	8	386	393	The SECLABEL of the user.
CDIR_UTK_EXECNODE	Char	8	395	402	The execution node of the work.
CDIR_UTK_SUSER_ID	Char	8	404	411	The submitting user ID.
CDIR_UTK_SNODE	Char	8	413	420	The submitting node.
CDIR_UTK_SGRP_ID	Char	8	422	429	The submitting group ID.
CDIR_UTK_SPOE	Char	8	431	438	The port of entry.
CDIR_UTK_SPCLASS	Char	8	440	447	Class of the POE. Valid values are "TERMINAL", "CONSOLE", "JESINPUT", and "APPCPORT".
CDIR_UTK_USER_ID	Char	8	449	456	User ID associated with the record.
CDIR_UTK_GRP_ID	Char	8	458	465	Group ID associated with the record.
CDIR_UTK_DFT_GRP	Yes/No	4	467	470	Is a default group assigned?
CDIR_UTK_DFT_SECL	Yes/No	4	472	475	Is a default SECLABEL assigned?
CDIR_APPC_LINK	Char	16	477	492	A key to link together audit records for a user's APPC transaction processing work.
CDIR_AUDIT_CODE	Char	11	494	504	Audit function code. For more information on the function codes, see Appendix I, "OpenExtensions Audit Function Codes," on page 439.
CDIR_OLD_REAL_UID	Integer	10	506	515	Old real UID.
CDIR_OLD_EFF_UID	Integer	10	517	526	Old effective UID.
CDIR_OLD_SAVED_UID	Integer	10	528	537	Old saved UID.
CDIR_OLD_REAL_GID	Integer	10	539	548	Old real GID.

Table 77. Format of the Change Directory Record Extension (continued)

Field Name	Type	Length	Position		Comments
			Start	End	
CDIR_OLD_EFF_GID	Integer	10	550	559	Old effective GID.
CDIR_OLD_SAVED_GID	Integer	10	561	570	Old saved GID.
CDIR_PATH_NAME	Char	1023	572	1594	The requested path name.
CDIR_FILE_ID	Char	32	1596	1627	File ID.
CDIR_FILE_OWN_UID	Integer	10	1629	1638	The owner UID associated with the file.
CDIR_FILE_OWN_GID	Integer	10	1640	1649	The owner GID associated with the file.
CDIR_DCE_LINK	Char	16	1651	1666	Link to connect DCE records that originate from a single DCE request.
CDIR_AUTH_TYPE	Char	13	1668	1680	Defines the type of request. Valid values are: "SERVER" and "AUTH_CLIENT" and "UNAUTH_CLIENT".

Event Qualifiers for Change Directory Requests

The event qualifiers that may be associated with a directory search event are shown in Table 78 on page 200.

Table 78. Event Code Qualifiers for Change Directory Records

Event Qualifier	Event Number	Event Description
SUCCESS	00	Current working directory changed. Failures are logged as directory search events.

The Format of the Change File Mode Record Extension

Table 79 on page 200 describes the format of a record that is created by changing the access mode of a file.

Table 79. Format of the Change File Mode Record Extension

Field Name	Type	Length	Position		Comments
			Start	End	
CMOD_CLASS	Char	8	282	289	Class name.
CMOD_USER_NAME	Char	20	291	310	The name associated with the user ID.
CMOD_UTK_ENCR	Yes/No	4	312	315	Is the UTOKEN associated with this user encrypted?
CMOD_UTK_PRE19	Yes/No	4	317	320	Is this a token for a release earlier than RACF 1.9?
CMOD_UTK_VERPROF	Yes/No	4	322	325	Is the VERIFYX propagation flag set?
CMOD_UTK_NJEUNUSR	Yes/No	4	327	330	Is this the NJE undefined user?
CMOD_UTK_LOGUSR	Yes/No	4	332	335	Is UAUDIT specified for this user?
CMOD_UTK_SPECIAL	Yes/No	4	337	340	Is this a RACF SPECIAL user?
CMOD_UTK_DEFAULT	Yes/No	4	342	345	Is this a default token?
CMOD_UTK_UNKNUSR	Yes/No	4	347	350	Is this an undefined user?
CMOD_UTK_ERROR	Yes/No	4	352	355	Is this user token in error?
CMOD_UTK_TRUSTED	Yes/No	4	357	360	Is this user a part of the trusted computing base (TCB)?
CMOD_UTK_SESSTYPE	Char	8	362	369	The session type of this session.

Table 79. Format of the Change File Mode Record Extension (continued)

Field Name	Type	Length	Position		Comments
			Start	End	
CMOD_UTK_SURROGAT	Yes/No	4	371	374	Is this a surrogate user?
CMOD_UTK_REMOTE	Yes/No	4	376	379	Is this a remote job?
CMOD_UTK_PRIV	Yes/No	4	381	384	Is this a privileged user ID?
CMOD_UTK_SECL	Char	8	386	393	The SECLABEL of the user.
CMOD_UTK_EXECNODE	Char	8	395	402	The execution node of the work.
CMOD_UTK_SUSER_ID	Char	8	404	411	The submitting user ID.
CMOD_UTK_SNODE	Char	8	413	420	The submitting node.
CMOD_UTK_SGRP_ID	Char	8	422	429	The submitting group ID.
CMOD_UTK_SPOE	Char	8	431	438	The port of entry.
CMOD_UTK_SPCLASS	Char	8	440	447	Class of the POE. Valid values are "TERMINAL", "CONSOLE", "JESINPUT", and "APPCPORT".
CMOD_UTK_USER_ID	Char	8	449	456	User ID associated with the record.
CMOD_UTK_GRP_ID	Char	8	458	465	Group ID associated with the record.
CMOD_UTK_DFT_GRP	Yes/No	4	467	470	Is a default group assigned?
CMOD_UTK_DFT_SECL	Yes/No	4	472	475	Is a default SECLABEL assigned?
CMOD_APPC_LINK	Char	16	477	492	Key to link together APPC records.
CMOD_AUDIT_CODE	Char	11	494	504	Audit function code. For more information on the function codes, see Appendix I, “OpenExtensions Audit Function Codes,” on page 439.
CMOD_OLD_REAL_UID	Integer	10	506	515	Old real UID.
CMOD_OLD_EFF_UID	Integer	10	517	526	Old effective UID.
CMOD_OLD_SAVED_UID	Integer	10	528	537	Old saved UID.
CMOD_OLD_REAL_GID	Integer	10	539	548	Old real GID.
CMOD_OLD_EFF_GID	Integer	10	550	559	Old effective GID.
CMOD_OLD_SAVED_GID	Integer	10	561	570	Old saved GID.
CMOD_PATH_NAME	Char	1023	572	1594	The requested path name.
CMOD_FILE_ID	Char	32	1596	1627	File ID.
CMOD_FILE_OWN_UID	Integer	10	1629	1638	The owner UID associated with the file.
CMOD_FILE_OWN_GID	Integer	10	1640	1649	The owner GID associated with the file.
CMOD_OLD_S_ISGID	Yes/No	4	1651	1654	Was the S_ISGID bit on for this file?
CMOD_OLD_S_ISUID	Yes/No	4	1656	1659	Was the S_ISUID bit on for this file?
CMOD_OLD_S_ISVTX	Yes/No	4	1661	1664	Was the S_ISVTX bit on for this file?
CMOD_OLD_OWN_READ	Yes/No	4	1666	1669	Was the owner READ bit on for this file?
CMOD_OLD_OWN_WRITE	Yes/No	4	1671	1674	Was the owner WRITE bit on for this file?
CMOD_OLD_OWN_EXEC	Yes/No	4	1676	1679	Was the owner EXECUTE bit on for this file?
CMOD_OLD_GRP_READ	Yes/No	4	1681	1684	Was the group READ bit on for this file?
CMOD_OLD_GRP_WRITE	Yes/No	4	1686	1689	Was the group WRITE bit on for this file?
CMOD_OLD_GRP_EXEC	Yes/No	4	1691	1694	Was the group EXECUTE bit on for this file?
CMOD_OLD_OTH_READ	Yes/No	4	1696	1699	Was the other READ bit on for this file?
CMOD_OLD_OTH_WRITE	Yes/No	4	1701	1704	Was the other WRITE bit on for this file?

Table 79. Format of the Change File Mode Record Extension (continued)

Field Name	Type	Length	Position		Comments
			Start	End	
CMOD_OLD_OTH_EXEC	Yes/No	4	1706	1709	Was the other EXECUTE bit on for this file?
CMOD_NEW_S_ISGID	Yes/No	4	1711	1714	Is the S_ISGID bit requested on for this file?
CMOD_NEW_S_ISUID	Yes/No	4	1716	1719	Is the S_ISUID bit requested on for this file?
CMOD_NEW_S_ISVTX	Yes/No	4	1721	1724	Is the S_ISVTX bit requested on for this file?
CMOD_NEW_OWN_READ	Yes/No	4	1726	1729	Is the owner READ bit on for this file?
CMOD_NEW_OWN_WRITE	Yes/No	4	1731	1734	Is the owner WRITE bit on for this file?
CMOD_NEW_OWN_EXEC	Yes/No	4	1736	1739	Is the owner EXECUTE bit on for this file?
CMOD_NEW_GRP_READ	Yes/No	4	1741	1744	Is the group READ bit on for this file?
CMOD_NEW_GRP_WRITE	Yes/No	4	1746	1749	Is the group WRITE bit on for this file?
CMOD_NEW_GRP_EXEC	Yes/No	4	1751	1754	Is the group EXECUTE bit on for this file?
CMOD_NEW_OTH_READ	Yes/No	4	1756	1759	Is the other READ bit on for this file?
CMOD_NEW_OTH_WRITE	Yes/No	4	1761	1764	Is the other WRITE bit on for this file?
CMOD_NEW_OTH_EXEC	Yes/No	4	1766	1769	Is the other EXECUTE bit on for this file?
CMOD_REQ_S_ISGID	Yes/No	4	1771	1774	Was the S_ISGID bit requested on for this file?
CMOD_REQ_S_ISUID	Yes/No	4	1776	1779	Was the S_ISUID bit requested on for this file?
CMOD_REQ_S_ISVTX	Yes/No	4	1781	1784	Was the S_ISVTX bit requested on for this file?
CMOD_REQ_OWN_READ	Yes/No	4	1786	1789	Was the owner READ bit requested on for this file?
CMOD_REQ_OWN_WRITE	Yes/No	4	1791	1794	Was the owner WRITE bit requested on for this file?
CMOD_REQ_OWN_EXEC	Yes/No	4	1796	1799	Was the owner EXECUTE bit requested on for this file?
CMOD_REQ_GRP_READ	Yes/No	4	1801	1804	Was the group READ bit requested on for this file?
CMOD_REQ_GRP_WRITE	Yes/No	4	1806	1809	Was the group WRITE bit requested on for this file?
CMOD_REQ_GRP_EXEC	Yes/No	4	1811	1814	Was the group EXECUTE bit requested on for this file?
CMOD_REQ_OTH_READ	Yes/No	4	1816	1819	Was the other READ bit requested on for this file?
CMOD_REQ_OTH_WRITE	Yes/No	4	1821	1824	Was the other WRITE bit requested on for this file?
CMOD_REQ_OTH_EXEC	Yes/No	4	1826	1829	Was the other EXECUTE bit requested on for this file?
CMOD_FILEPOOL	Char	8	1831	1838	SFS filepool containing the BFS file.
CMOD_FILESPACE	Char	8	1840	1847	SFS filespace containing the BFS file.
CMOD_INODE	Integer	10	1849	1858	Inode (file serial number).
CMOD_SCID	Integer	10	1860	1869	File SCID.
CMOD_DCE_LINK	Char	16	1871	1886	Link to connect DCE records that originate from a single DCE request.
CMOD_AUTH_TYPE	Char	13	1888	1900	Defines the type of request. Valid values are: "SERVER" and "AUTH_CLIENT" and "UNAUTH_CLIENT".

Event Qualifiers for Change File Mode Records

The event qualifiers that may be associated with changing a file mode event are shown in Table 80 on page 203.

Table 80. Event Code Qualifiers for Change File Mode Records		
Event Qualifier	**Event Number**	**Event Description**
SUCCESS	00	File's mode changed.
NOTAUTH	01	Not authorized to change the file's mode.

The Format of the Change File Ownership Record Extension

Table 81 on page 203 describes the format of a record that is created by changing the ownership of a file.

Table 81. Format of the Change File Ownership Record Extension

Field Name	Type	Length	Position		Comments
			Start	End	
COWN_CLASS	Char	8	282	289	Class name.
COWN_USER_NAME	Char	20	291	310	The name associated with the user ID.
COWN_UTK_ENCR	Yes/No	4	312	315	Is the UTOKEN associated with this user encrypted?
COWN_UTK_PRE19	Yes/No	4	317	320	Is this a token for a release earlier than RACF 1.9?
COWN_UTK_VERPROF	Yes/No	4	322	325	Is the VERIFYX propagation flag set?
COWN_UTK_NJEUNUSR	Yes/No	4	327	330	Is this the NJE undefined user?
COWN_UTK_LOGUSR	Yes/No	4	332	335	Is UAUDIT specified for this user?
COWN_UTK_SPECIAL	Yes/No	4	337	340	Is this a RACF SPECIAL user?
COWN_UTK_DEFAULT	Yes/No	4	342	345	Is this a default token?
COWN_UTK_UNKNUSR	Yes/No	4	347	350	Is this an undefined user?
COWN_UTK_ERROR	Yes/No	4	352	355	Is this user token in error?
COWN_UTK_TRUSTED	Yes/No	4	357	360	Is this user a part of the trusted computing base (TCB)?
COWN_UTK_SESSTYPE	Char	8	362	369	The session type of this session.
COWN_UTK_SURROGAT	Yes/No	4	371	374	Is this a surrogate user?
COWN_UTK_REMOTE	Yes/No	4	376	379	Is this a remote job?
COWN_UTK_PRIV	Yes/No	4	381	384	Is this a privileged user ID?
COWN_UTK_SECL	Char	8	386	393	The SECLABEL of the user.
COWN_UTK_EXECNODE	Char	8	395	402	The execution node of the work.
COWN_UTK_SUSER_ID	Char	8	404	411	The submitting user ID.
COWN_UTK_SNODE	Char	8	413	420	The submitting node.
COWN_UTK_SGRP_ID	Char	8	422	429	The submitting group ID.
COWN_UTK_SPOE	Char	8	431	438	The port of entry.
COWN_UTK_SPCLASS	Char	8	440	447	Class of the POE. Valid values are "TERMINAL", "CONSOLE", "JESINPUT", and "APPCPORT".
COWN_UTK_USER_ID	Char	8	449	456	User ID associated with the record.
COWN_UTK_GRP_ID	Char	8	458	465	Group ID associated with the record.
COWN_UTK_DFT_GRP	Yes/No	4	467	470	Is a default group assigned?
COWN_UTK_DFT_SECL	Yes/No	4	472	475	Is a default SECLABEL assigned?
COWN_APPC_LINK	Char	16	477	492	Key to link together APPC records.

Table 81. Format of the Change File Ownership Record Extension (continued)

Field Name	Type	Length	Position		Comments
			Start	End	
COWN_AUDIT_CODE	Char	11	494	504	Audit function code. For more information on the function codes, see Appendix I, "OpenExtensions Audit Function Codes," on page 439.
COWN_OLD_REAL_UID	Integer	10	506	515	Old real UID.
COWN_OLD_EFF_UID	Integer	10	517	526	Old effective UID.
COWN_OLD_SAVED_UID	Integer	10	528	537	Old saved UID.
COWN_OLD_REAL_GID	Integer	10	539	548	Old real GID.
COWN_OLD_EFF_GID	Integer	10	550	559	Old effective GID.
COWN_OLD_SAVED_GID	Integer	10	561	570	Old saved GID.
COWN_PATH_NAME	Char	1023	572	1594	The requested path name.
COWN_FILE_ID	Char	32	1596	1627	File ID.
COWN_FILE_OWN_UID	Integer	10	1629	1638	The owner UID associated with the file.
COWN_FILE_OWN_GID	Integer	10	1640	1649	The owner GID associated with the file.
COWN_UID	Integer	10	1651	1660	The UID input parameter.
COWN_GID	Integer	10	1662	1671	The GID input parameter.
COWN_FILEPOOL	Char	8	1673	1680	SFS filepool containing the BFS file.
COWN_FILESPACE	Char	8	1682	1689	SFS filespace containing the BFS file.
COWN_INODE	Integer	10	1691	1700	Inode (file serial number).
COWN_SCID	Integer	10	1702	1711	File SCID.
COWN_DCE_LINK	Char	16	1713	1728	Link to connect DCE records that originate from a single DCE request.
COWN_AUTH_TYPE	Char	13	1730	1742	Defines the type of request. Valid values are: "SERVER" and "AUTH_CLIENT" and "UNAUTH_CLIENT".

Event Qualifiers for Change File Ownership Requests

The event qualifiers that may be associated with changing a file's ownership are shown in Table 82 on page 204.

Table 82. Event Code Qualifiers for Change File Ownership Records

Event Qualifier	Event Number	Event Description
SUCCESS	00	File's ownership changed.
NOTAUTH	01	Not authorized to change the file's ownership.

The Format of the Clear SETID Bits Record Extension

Table 83 on page 204 describes the format of a record that is created by clearing the SETID bits of a file.

Table 83. Format of the Clear SETID Bits Record Extension

Field Name	Type	Length	Position		Comments
			Start	End	
CSID_CLASS	Char	8	282	289	Class name.
CSID_USER_NAME	Char	20	291	310	The name associated with the user ID.

Table 83. Format of the Clear SETID Bits Record Extension (continued)

Field Name	Type	Length	Position		Comments
			Start	End	
CSID_UTK_ENCR	Yes/No	4	312	315	Is the UTOKEN associated with this user encrypted?
CSID_UTK_PRE19	Yes/No	4	317	320	Is this a token for a release earlier than RACF 1.9?
CSID_UTK_VERPROF	Yes/No	4	322	325	Is the VERIFYX propagation flag set?
CSID_UTK_NJEUNUSR	Yes/No	4	327	330	Is this the NJE undefined user?
CSID_UTK_LOGUSR	Yes/No	4	332	335	Is UAUDIT specified for this user?
CSID_UTK_SPECIAL	Yes/No	4	337	340	Is this a RACF SPECIAL user?
CSID_UTK_DEFAULT	Yes/No	4	342	345	Is this a default token?
CSID_UTK_UNKNUSR	Yes/No	4	347	350	Is this an undefined user?
CSID_UTK_ERROR	Yes/No	4	352	355	Is this user token in error?
CSID_UTK_TRUSTED	Yes/No	4	357	360	Is this user a part of the trusted computing base (TCB)?
CSID_UTK_SESSTYPE	Char	8	362	369	The session type of this session.
CSID_UTK_SURROGAT	Yes/No	4	371	374	Is this a surrogate user?
CSID_UTK_REMOTE	Yes/No	4	376	379	Is this a remote job?
CSID_UTK_PRIV	Yes/No	4	381	384	Is this a privileged user ID?
CSID_UTK_SECL	Char	8	386	393	The SECLABEL of the user.
CSID_UTK_EXECNODE	Char	8	395	402	The execution node of the work.
CSID_UTK_SUSER_ID	Char	8	404	411	The submitting user ID.
CSID_UTK_SNODE	Char	8	413	420	The submitting node.
CSID_UTK_SGRP_ID	Char	8	422	429	The submitting group ID.
CSID_UTK_SPOE	Char	8	431	438	The port of entry.
CSID_UTK_SPCLASS	Char	8	440	447	Class of the POE. Valid values are "TERMINAL", "CONSOLE", "JESINPUT", and "APPCPORT".
CSID_UTK_USER_ID	Char	8	449	456	User ID associated with the record.
CSID_UTK_GRP_ID	Char	8	458	465	Group ID associated with the record.
CSID_UTK_DFT_GRP	Yes/No	4	467	470	Is a default group assigned?
CSID_UTK_DFT_SECL	Yes/No	4	472	475	Is a default SECLABEL assigned?
CSID_APPC_LINK	Char	16	477	492	A key to link together audit records for a user's APPC transaction processing work.
CSID_AUDIT_CODE	Char	11	494	504	Audit function code. For more information on the function codes, see Appendix I, “OpenExtensions Audit Function Codes,” on page 439.
CSID_OLD_REAL_UID	Integer	10	506	515	Old real UID.
CSID_OLD_EFF_UID	Integer	10	517	526	Old effective UID.
CSID_OLD_SAVED_UID	Integer	10	528	537	Old saved UID.
CSID_OLD_REAL_GID	Integer	10	539	548	Old real GID.
CSID_OLD_EFF_GID	Integer	10	550	559	Old effective GID.
CSID_OLD_SAVED_GID	Integer	10	561	570	Old saved GID.
CSID_PATH_NAME	Char	1023	572	1594	The requested path name.
CSID_FILE_ID	Char	32	1596	1627	File ID.
CSID_FILE_OWN_UID	Integer	10	1629	1638	The owner UID associated with the file.

Table 83. Format of the Clear SETID Bits Record Extension (continued)

Field Name	Type	Length	Position		Comments
			Start	End	
CSID_FILE_OWN_GID	Integer	10	1640	1649	The owner GID associated with the file.
CSID_OLD_S_ISGID	Yes/No	4	1651	1654	Was the S_ISGID bit requested on for this file?
CSID_OLD_S_ISUID	Yes/No	4	1656	1659	Was the S_ISUID bit requested on for this file?
CSID_OLD_S_ISVTX	Yes/No	4	1661	1664	Was the S_ISVTX bit requested on for this file?
CSID_OLD_OWN_READ	Yes/No	4	1666	1669	Was the owner READ bit on for this file?
CSID_OLD_OWN_WRITE	Yes/No	4	1671	1674	Was the owner WRITE bit on for this file?
CSID_OLD_OWN_EXEC	Yes/No	4	1676	1679	Was the owner EXECUTE bit on for this file?
CSID_OLD_GRP_READ	Yes/No	4	1681	1684	Was the group READ bit on for this file?
CSID_OLD_GRP_WRITE	Yes/No	4	1686	1689	Was the group WRITE bit on for this file?
CSID_OLD_GRP_EXEC	Yes/No	4	1691	1694	Was the group EXECUTE bit on for this file?
CSID_OLD_OTH_READ	Yes/No	4	1696	1699	Was the other READ bit on for this file?
CSID_OLD_OTH_WRITE	Yes/No	4	1701	1704	Was the other WRITE bit on for this file?
CSID_OLD_OTH_EXEC	Yes/No	4	1706	1709	Was the other EXECUTE bit on for this file?
CSID_NEW_S_ISGID	Yes/No	4	1711	1714	Is the S_ISGID bit requested on for this file?
CSID_NEW_S_ISUID	Yes/No	4	1716	1719	Is the S_ISUID bit requested on for this file?
CSID_NEW_S_ISVTX	Yes/No	4	1721	1724	Is the S_ISVTX bit requested on for this file?
CSID_NEW_OWN_READ	Yes/No	4	1726	1729	Is the owner READ bit on for this file?
CSID_NEW_OWN_WRITE	Yes/No	4	1731	1734	Is the owner WRITE bit on for this file?
CSID_NEW_OWN_EXEC	Yes/No	4	1736	1739	Is the owner EXECUTE bit on for this file?
CSID_NEW_GRP_READ	Yes/No	4	1741	1744	Is the group READ bit on for this file?
CSID_NEW_GRP_WRITE	Yes/No	4	1746	1749	Is the group WRITE bit on for this file?
CSID_NEW_GRP_EXEC	Yes/No	4	1751	1754	Is the group EXECUTE bit on for this file?
CSID_NEW_OTH_READ	Yes/No	4	1756	1759	Is the other READ bit on for this file?
CSID_NEW_OTH_WRITE	Yes/No	4	1761	1764	Is the other WRITE bit on for this file?
CSID_NEW_OTH_EXEC	Yes/No	4	1766	1769	Is the other EXECUTE bit on for this file?

Event Qualifiers for Clear SETID Requests

The event qualifier that may be associated with clearing a file's SETID bits is shown in Table 84 on page 206.

Table 84. Event Code Qualifiers for Clear SETID Records

Event Qualifier	Event Number	Event Description
SUCCESS	00	S_ISUID, S_ISGID. and S_ISVTX changed. There are no failure cases for this event.

The Format of the EXEC SETUID/SETGID Record Extension

Table 85 on page 207 describes the format of a record that is created by the execution of an EXEC SETUID or SETGID.

Table 85. Format of the EXEC with SETUID/SETGID Record Extension

Field Name	Type	Length	Position		Comments
			Start	End	
ESID_CLASS	Char	8	282	289	Class name.
ESID_USER_NAME	Char	20	291	310	The name associated with the user ID.
ESID_UTK_ENCR	Yes/No	4	312	315	Is the UTOKEN associated with this user encrypted?
ESID_UTK_PRE19	Yes/No	4	317	320	Is this a token for a release earlier than RACF 1.9?
ESID_UTK_VERPROF	Yes/No	4	322	325	Is the VERIFYX propagation flag set?
ESID_UTK_NJEUNUSR	Yes/No	4	327	330	Is this the NJE undefined user?
ESID_UTK_LOGUSR	Yes/No	4	332	335	Is UAUDIT specified for this user?
ESID_UTK_SPECIAL	Yes/No	4	337	340	Is this a RACF SPECIAL user?
ESID_UTK_DEFAULT	Yes/No	4	342	345	Is this a default token?
ESID_UTK_UNKNUSR	Yes/No	4	347	350	Is this an undefined user?
ESID_UTK_ERROR	Yes/No	4	352	355	Is this user token in error?
ESID_UTK_TRUSTED	Yes/No	4	357	360	Is this user a part of the trusted computing base (TCB)?
ESID_UTK_SESSTYPE	Char	8	362	369	The session type of this session.
ESID_UTK_SURROGAT	Yes/No	4	371	374	Is this a surrogate user?
ESID_UTK_REMOTE	Yes/No	4	376	379	Is this a remote job?
ESID_UTK_PRIV	Yes/No	4	381	384	Is this a privileged user ID?
ESID_UTK_SECL	Char	8	386	393	The SECLABEL of the user.
ESID_UTK_EXECNODE	Char	8	395	402	The execution node of the work.
ESID_UTK_SUSER_ID	Char	8	404	411	The submitting user ID.
ESID_UTK_SNODE	Char	8	413	420	The submitting node.
ESID_UTK_SGRP_ID	Char	8	422	429	The submitting group ID.
ESID_UTK_SPOE	Char	8	431	438	The port of entry.
ESID_UTK_SPCLASS	Char	8	440	447	Class of the POE. Valid values are "TERMINAL", "CONSOLE", "JESINPUT", and "APPCPORT".
ESID_UTK_USER_ID	Char	8	449	456	User ID associated with the record.
ESID_UTK_GRP_ID	Char	8	458	465	Group ID associated with the record.
ESID_UTK_DFT_GRP	Yes/No	4	467	470	Is a default group assigned?
ESID_UTK_DFT_SECL	Yes/No	4	472	475	Is a default SECLABEL assigned?
ESID_APPC_LINK	Char	16	477	492	A key to link together audit records for a user's APPC transaction processing work.
ESID_AUDIT_CODE	Char	11	494	504	Audit function code. For more information on the function codes, see Appendix I, “OpenExtensions Audit Function Codes,” on page 439.
ESID_OLD_REAL_UID	Integer	10	506	515	Old real UID.
ESID_OLD_EFF_UID	Integer	10	517	526	Old effective UID.
ESID_OLD_SAVED_UID	Integer	10	528	537	Old saved UID.
ESID_OLD_REAL_GID	Integer	10	539	548	Old real GID.
ESID_OLD_EFF_GID	Integer	10	550	559	Old effective GID.
ESID_OLD_SAVED_GID	Integer	10	561	570	Old saved GID.
ESID_NEW_REAL_UID	Integer	10	572	581	New real UID.

Table 85. Format of the EXEC with SETUID/SETGID Record Extension (continued)					
Field Name	**Type**	**Length**	**Position**		**Comments**
			Start	**End**	
ESID_NEW_EFF_UID	Integer	10	583	592	New effective UID.
ESID_NEW_SAVED_UID	Integer	10	594	603	New saved UID.
ESID_NEW_REAL_GID	Integer	10	605	614	New real GID.
ESID_NEW_EFF_GID	Integer	10	616	625	New effective GID.
ESID_NEW_SAVED_GID	Integer	10	627	636	New saved GID.
ESID_UID	Integer	10	638	647	The UID input parameter.
ESID_GID	Integer	10	649	658	The GID input parameter.

Event Qualifiers for EXEC with SETUID/SETGID Record Extension

The event qualifier that may be associated with the execution of EXEC SETUID or EXEC SETGID is shown in Table 86 on page 208.

Table 86. Event Code Qualifiers for EXEC with SETUID/SETGID Records		
Event Qualifier	**Event Number**	**Event Description**
SUCCESS	00	UID or GID changed. There are no failure cases for this event on z/OS.
NOUAUTH	01	User not authorized to EXEC.U*uid* profile in the VMPOSIX class (z/VM only)
NOGAUTH	02	User not authorized to EXEC.G*gid* profile in the VMPOSIX class (z/VM only)

The Format of the GETPSENT Record Extension

Table 87 on page 208 describes the format of a record that is created by the GETPSENT service.

Table 87. Format of the GETPSENT Record Extension					
Field Name	**Type**	**Length**	**Position**		**Comments**
			Start	**End**	
GPST_CLASS	Char	8	282	289	Class name.
GPST_USER_NAME	Char	20	291	310	The name associated with the user ID.
GPST_UTK_ENCR	Yes/No	4	312	315	Is the UTOKEN associated with this user encrypted?
GPST_UTK_PRE19	Yes/No	4	317	320	Is this a token for a release earlier than RACF 1.9?
GPST_UTK_VERPROF	Yes/No	4	322	325	Is the VERIFYX propagation flag set?
GPST_UTK_NJEUNUSR	Yes/No	4	327	330	Is this the NJE undefined user?
GPST_UTK_LOGUSR	Yes/No	4	332	335	Is UAUDIT specified for this user?
GPST_UTK_SPECIAL	Yes/No	4	337	340	Is this a RACF SPECIAL user?
GPST_UTK_DEFAULT	Yes/No	4	342	345	Is this a default token?
GPST_UTK_UNKNUSR	Yes/No	4	347	350	Is this an undefined user?
GPST_UTK_ERROR	Yes/No	4	352	355	Is this user token in error?
GPST_UTK_TRUSTED	Yes/No	4	357	360	Is this user a part of the trusted computing base (TCB)?
GPST_UTK_SESSTYPE	Char	8	362	369	The session type of this session.
GPST_UTK_SURROGAT	Yes/No	4	371	374	Is this a surrogate user?
GPST_UTK_REMOTE	Yes/No	4	376	379	Is this a remote job?

Table 87. Format of the GETPSENT Record Extension (continued)

Field Name	Type	Length	Position		Comments
			Start	End	
GPST_UTK_PRIV	Yes/No	4	381	384	Is this a privileged user ID?
GPST_UTK_SECL	Char	8	386	393	The SECLABEL of the user.
GPST_UTK_EXECNODE	Char	8	395	402	The execution node of the work.
GPST_UTK_SUSER_ID	Char	8	404	411	The submitting user ID.
GPST_UTK_SNODE	Char	8	413	420	The submitting node.
GPST_UTK_SGRP_ID	Char	8	422	429	The submitting group ID.
GPST_UTK_SPOE	Char	8	431	438	The port of entry.
GPST_UTK_SPCLASS	Char	8	440	447	Class of the POE. Valid values are "TERMINAL", "CONSOLE", "JESINPUT", and "APPCPORT".
GPST_UTK_USER_ID	Char	8	449	456	User ID associated with the record.
GPST_UTK_GRP_ID	Char	8	458	465	Group ID associated with the record.
GPST_UTK_DFT_GRP	Yes/No	4	467	470	Is a default group assigned?
GPST_UTK_DFT_SECL	Yes/No	4	472	475	Is a default SECLABEL assigned?
GPST_APPC_LINK	Char	16	477	492	Key to link together APPC records.
GPST_AUDIT_CODE	Char	11	494	504	Audit function code. For more information on the function codes, see Appendix I, “OpenExtensions Audit Function Codes,” on page 439.
GPST_OLD_REAL_UID	Integer	10	506	515	Old real UID.
GPST_OLD_EFF_UID	Integer	10	517	526	Old effective UID.
GPST_OLD_SAVED_UID	Integer	10	528	537	Old saved UID.
GPST_OLD_REAL_GID	Integer	10	539	548	Old real GID.
GPST_OLD_EFF_GID	Integer	10	550	559	Old effective GID.
GPST_OLD_SAVED_GID	Integer	10	561	570	Old saved GID.
GPST_TGT_REAL_UID	Integer	10	572	581	Target real UID.
GPST_TGT_EFF_UID	Integer	10	583	592	Target effective UID.
GPST_TGT_SAV_UID	Integer	10	594	603	Target saved UID.
GPST_TGT_PID	Integer	10	605	614	Target process ID.

Event Qualifiers for the GETPSENT Record Extension

The event qualifiers that may be associated with the GETPSENT service are shown in Table 88 on page 209.

Table 88. Event Code Qualifiers for GETPSENT Records

Event Qualifier	Event Number	Event Description
SUCCESS	00	GETPSENT was successful.
NOTAUTH	01	Not authorized to the specified process.

The Format of the Initialize OpenExtensions Process Record

Table 89 on page 210 describes the format of a record that is created when an OpenExtensions process is initialized.

Table 89. Format of the Initialize OpenExtensions Process Record Extension

Field Name	Type	Length	Position		Comments
			Start	End	
IOEP_CLASS	Char	8	282	289	Class name.
IOEP_USER_NAME	Char	20	291	310	The name associated with the user ID.
IOEP_UTK_ENCR	Yes/No	4	312	315	Is the UTOKEN associated with this user encrypted?
IOEP_UTK_PRE19	Yes/No	4	317	320	Is this a token for a release earlier than RACF 1.9?
IOEP_UTK_VERPROF	Yes/No	4	322	325	Is the VERIFYX propagation flag set?
IOEP_UTK_NJEUNUSR	Yes/No	4	327	330	Is this the NJE undefined user?
IOEP_UTK_LOGUSR	Yes/No	4	332	335	Is UAUDIT specified for this user?
IOEP_UTK_SPECIAL	Yes/No	4	337	340	Is this a RACF SPECIAL user?
IOEP_UTK_DEFAULT	Yes/No	4	342	345	Is this a default token?
IOEP_UTK_UNKNUSR	Yes/No	4	347	350	Is this an undefined user?
IOEP_UTK_ERROR	Yes/No	4	352	355	Is this user token in error?
IOEP_UTK_TRUSTED	Yes/No	4	357	360	Is this user a part of the trusted computing base (TCB)?
IOEP_UTK_SESSTYPE	Char	8	362	369	The session type of this session.
IOEP_UTK_SURROGAT	Yes/No	4	371	374	Is this a surrogate user?
IOEP_UTK_REMOTE	Yes/No	4	376	379	Is this a remote job?
IOEP_UTK_PRIV	Yes/No	4	381	384	Is this a privileged user ID?
IOEP_UTK_SECL	Char	8	386	393	The SECLABEL of the user.
IOEP_UTK_EXECNODE	Char	8	395	402	The execution node of the work.
IOEP_UTK_SUSER_ID	Char	8	404	411	The submitting user ID.
IOEP_UTK_SNODE	Char	8	413	420	The submitting node.
IOEP_UTK_SGRP_ID	Char	8	422	429	The submitting group ID.
IOEP_UTK_SPOE	Char	8	431	438	The port of entry.
IOEP_UTK_SPCLASS	Char	8	440	447	Class of the POE. Valid values are "TERMINAL", "CONSOLE", "JESINPUT", and "APPCPORT".
IOEP_UTK_USER_ID	Char	8	449	456	User ID associated with the record.
IOEP_UTK_GRP_ID	Char	8	458	465	Group ID associated with the record.
IOEP_UTK_DFT_GRP	Yes/No	4	467	470	Is a default group assigned?
IOEP_UTK_DFT_SECL	Yes/No	4	472	475	Is a default SECLABEL assigned?
IOEP_APPC_LINK	Char	16	477	492	A key to link together audit records for a user's APPC transaction processing work.
IOEP_AUDIT_CODE	Char	11	494	504	Audit function code. For more information on the function codes, see Appendix I, “OpenExtensions Audit Function Codes,” on page 439.
IOEP_OLD_REAL_UID	Integer	10	506	515	Old real UID.
IOEP_OLD_EFF_UID	Integer	10	517	526	Old effective UID.
IOEP_OLD_SAVED_UID	Integer	10	528	537	Old saved UID.
IOEP_OLD_REAL_GID	Integer	10	539	548	Old real GID.
IOEP_OLD_EFF_GID	Integer	10	550	559	Old effective GID.
IOEP_OLD_SAVED_GID	Integer	10	561	570	Old saved GID.

Event Qualifiers for the Initialize OpenExtensions Process Records

The event qualifiers that may be associated with the initiation of an OpenExtensions process are shown in Table 90 on page 211.

Table 90. Event Code Qualifiers for Initialize OpenExtensions Process Records

Event Qualifier	Event Number	Event Description
SUCCESS	00	Process successfully initialized.
NOTDFND	01	User not defined as an OpenExtensions user. The OpenExtensions segment or the user profile was missing.
NOUID	02	Incompletely defined user ID. There was no UID in profile.
NOGID	03.	User's current group has no GID.

The Format of the OpenExtensions Process Completion Record

Table 91 on page 211 describes the format of a record that is created when an OpenExtensions process completes.

Table 91. Format of the OpenExtensions Process Completion Record Extension

Field Name	Type	Length	Position		Comments
			Start	End	
TOEP_CLASS	Char	8	282	289	Class name.
TOEP_USER_NAME	Char	20	291	310	The name associated with the user ID.
TOEP_UTK_ENCR	Yes/No	4	312	315	Is the UTOKEN associated with this user encrypted?
TOEP_UTK_PRE19	Yes/No	4	317	320	Is this a token for a release earlier than RACF 1.9?
TOEP_UTK_VERPROF	Yes/No	4	322	325	Is the VERIFYX propagation flag set?
TOEP_UTK_NJEUNUSR	Yes/No	4	327	330	Is this the NJE undefined user?
TOEP_UTK_LOGUSR	Yes/No	4	332	335	Is UAUDIT specified for this user?
TOEP_UTK_SPECIAL	Yes/No	4	337	340	Is this a RACF SPECIAL user?
TOEP_UTK_DEFAULT	Yes/No	4	342	345	Is this a default token?
TOEP_UTK_UNKNUSR	Yes/No	4	347	350	Is this an undefined user?
TOEP_UTK_ERROR	Yes/No	4	352	355	Is this user token in error?
TOEP_UTK_TRUSTED	Yes/No	4	357	360	Is this user a part of the trusted computing base (TCB)?
TOEP_UTK_SESSTYPE	Char	8	362	369	The session type of this session.
TOEP_UTK_SURROGAT	Yes/No	4	371	374	Is this a surrogate user?
TOEP_UTK_REMOTE	Yes/No	4	376	379	Is this a remote job?
TOEP_UTK_PRIV	Yes/No	4	381	384	Is this a privileged user ID?
TOEP_UTK_SECL	Char	8	386	393	The SECLABEL of the user.
TOEP_UTK_EXECNODE	Char	8	395	402	The execution node of the work.
TOEP_UTK_SUSER_ID	Char	8	404	411	The submitting user ID.
TOEP_UTK_SNODE	Char	8	413	420	The submitting node.
TOEP_UTK_SGRP_ID	Char	8	422	429	The submitting group ID.
TOEP_UTK_SPOE	Char	8	431	438	The port of entry.
TOEP_UTK_SPCLASS	Char	8	440	447	Class of the POE. Valid values are "TERMINAL", "CONSOLE", "JESINPUT", and "APPCPORT".

Table 91. Format of the OpenExtensions Process Completion Record Extension (continued)

Field Name	Type	Length	Position		Comments
			Start	End	
TOEP_UTK_USER_ID	Char	8	449	456	User ID associated with the record.
TOEP_UTK_GRP_ID	Char	8	458	465	Group ID associated with the record.
TOEP_UTK_DFT_GRP	Yes/No	4	467	470	Is a default group assigned?
TOEP_UTK_DFT_SECL	Yes/No	4	472	475	Is a default SECLABEL assigned?
TOEP_APPC_LINK	Char	16	477	492	A key to link together audit records for a user's APPC transaction processing work.
TOEP_AUDIT_CODE	Char	11	494	504	Audit function code. For more information on the function codes, see Appendix I, "OpenExtensions Audit Function Codes," on page 439.
TOEP_OLD_REAL_UID	Integer	10	506	515	Old real UID.
TOEP_OLD_EFF_UID	Integer	10	517	526	Old effective UID.
TOEP_OLD_SAVED_UID	Integer	10	528	537	Old saved UID.
TOEP_OLD_REAL_GID	Integer	10	539	548	Old real GID.
TOEP_OLD_EFF_GID	Integer	10	550	559	Old effective GID.
TOEP_OLD_SAVED_GID	Integer	10	561	570	Old saved GID.

Event Qualifiers for the OpenExtensions Process Completion Record

The event qualifier that may be associated with the completion of an OpenExtensions process is shown in Table 92 on page 212.

Table 92. Event Code Qualifiers for OpenExtensions Process Completion Records

Event Qualifier	Event Number	Event Description
SUCCESS	00	Process complete. There are no failure cases for this event.

The Format of the KILL Process Record Extension

Table 93 on page 212 describes the format of a record that is created by the termination with extreme prejudice of a process.

Table 93. Format of the KILL Process Record Extension

Field Name	Type	Length	Position		Comments
			Start	End	
KILL_CLASS	Char	8	282	289	Class name.
KILL_USER_NAME	Char	20	291	310	The name associated with the user ID.
KILL_UTK_ENCR	Yes/No	4	312	315	Is the UTOKEN associated with this user encrypted?
KILL_UTK_PRE19	Yes/No	4	317	320	Is this a token for a release earlier than RACF 1.9?
KILL_UTK_VERPROF	Yes/No	4	322	325	Is the VERIFYX propagation flag set?
KILL_UTK_NJEUNUSR	Yes/No	4	327	330	Is this the NJE undefined user?
KILL_UTK_LOGUSR	Yes/No	4	332	335	Is UAUDIT specified for this user?
KILL_UTK_SPECIAL	Yes/No	4	337	340	Is this a RACF SPECIAL user?
KILL_UTK_DEFAULT	Yes/No	4	342	345	Is this a default token?
KILL_UTK_UNKNUSR	Yes/No	4	347	350	Is this an undefined user?

Table 93. Format of the KILL Process Record Extension (continued)

Field Name	Type	Length	Position		Comments
			Start	End	
KILL_UTK_ERROR	Yes/No	4	352	355	Is this user token in error?
KILL_UTK_TRUSTED	Yes/No	4	357	360	Is this user a part of the trusted computing base (TCB)?
KILL_UTK_SESSTYPE	Char	8	362	369	The session type of this session.
KILL_UTK_SURROGAT	Yes/No	4	371	374	Is this a surrogate user?
KILL_UTK_REMOTE	Yes/No	4	376	379	Is this a remote job?
KILL_UTK_PRIV	Yes/No	4	381	384	Is this a privileged user ID?
KILL_UTK_SECL	Char	8	386	393	The SECLABEL of the user.
KILL_UTK_EXECNODE	Char	8	395	402	The execution node of the work.
KILL_UTK_SUSER_ID	Char	8	404	411	The submitting user ID.
KILL_UTK_SNODE	Char	8	413	420	The submitting node.
KILL_UTK_SGRP_ID	Char	8	422	429	The submitting group ID.
KILL_UTK_SPOE	Char	8	431	438	The port of entry.
KILL_UTK_SPCLASS	Char	8	440	447	Class of the POE. Valid values are "TERMINAL", "CONSOLE", "JESINPUT", and "APPCPORT".
KILL_UTK_USER_ID	Char	8	449	456	User ID associated with the record.
KILL_UTK_GRP_ID	Char	8	458	465	Group ID associated with the record.
KILL_UTK_DFT_GRP	Yes/No	4	467	470	Is a default group assigned?
KILL_UTK_DFT_SECL	Yes/No	4	472	475	Is a default SECLABEL assigned?
KILL_APPC_LINK	Char	16	477	492	Key to link together APPC records.
KILL_AUDIT_CODE	Char	11	494	504	Audit function code. For more information on the function codes, see Appendix I, “OpenExtensions Audit Function Codes,” on page 439.
KILL_OLD_REAL_UID	Integer	10	506	515	Old real UID.
KILL_OLD_EFF_UID	Integer	10	517	526	Old effective UID.
KILL_OLD_SAVED_UID	Integer	10	528	537	Old saved UID.
KILL_OLD_REAL_GID	Integer	10	539	548	Old real GID.
KILL_OLD_EFF_GID	Integer	10	550	559	Old effective GID.
KILL_OLD_SAVED_GID	Integer	10	561	570	Old saved GID.
KILL_TGT_REAL_UID	Integer	10	572	581	Target real UID.
KILL_TGT_EFF_UID	Integer	10	583	592	Target effective UID.
KILL_TGT_SAV_UID	Integer	10	594	603	Target saved UID.
KILL_TGT_PID	Integer	10	605	614	Target process ID.
KILL_SIGNAL_CODE	Integer	10	616	625	Kill signal code.

Event Qualifiers for the KILL Process Record Extension

The event qualifiers that may be associated with the killing of a process are shown in Table 94 on page 214.

Table 94. Event Code Qualifiers for KILL Process Records		
Event Qualifier	**Event Number**	**Event Description**
SUCCESS	00	Process terminated.
NOTAUTH	01	Not authorized to kill the specified process.

The Format of the LINK Record Extension

Table 95 on page 214 describes the format of a record that is created by a LINK operation.

Table 95. Format of the LINK Record Extension					
Field Name	**Type**	**Length**	**Position**		**Comments**
			Start	**End**	
LINK_CLASS	Char	8	282	289	Class name.
LINK_USER_NAME	Char	20	291	310	The name associated with the user ID.
LINK_UTK_ENCR	Yes/No	4	312	315	Is the UTOKEN associated with this user encrypted?
LINK_UTK_PRE19	Yes/No	4	317	320	Is this a token for a release earlier than RACF 1.9?
LINK_UTK_VERPROF	Yes/No	4	322	325	Is the VERIFYX propagation flag set?
LINK_UTK_NJEUNUSR	Yes/No	4	327	330	Is this the NJE undefined user?
LINK_UTK_LOGUSR	Yes/No	4	332	335	Is UAUDIT specified for this user?
LINK_UTK_SPECIAL	Yes/No	4	337	340	Is this a RACF SPECIAL user?
LINK_UTK_DEFAULT	Yes/No	4	342	345	Is this a default token?
LINK_UTK_UNKNUSR	Yes/No	4	347	350	Is this an undefined user?
LINK_UTK_ERROR	Yes/No	4	352	355	Is this user token in error?
LINK_UTK_TRUSTED	Yes/No	4	357	360	Is this user a part of the trusted computing base (TCB)?
LINK_UTK_SESSTYPE	Char	8	362	369	The session type of this session.
LINK_UTK_SURROGAT	Yes/No	4	371	374	Is this a surrogate user?
LINK_UTK_REMOTE	Yes/No	4	376	379	Is this a remote job?
LINK_UTK_PRIV	Yes/No	4	381	384	Is this a privileged user ID?
LINK_UTK_SECL	Char	8	386	393	The SECLABEL of the user.
LINK_UTK_EXECNODE	Char	8	395	402	The execution node of the work.
LINK_UTK_SUSER_ID	Char	8	404	411	The submitting user ID.
LINK_UTK_SNODE	Char	8	413	420	The submitting node.
LINK_UTK_SGRP_ID	Char	8	422	429	The submitting group ID.
LINK_UTK_SPOE	Char	8	431	438	The port of entry.
LINK_UTK_SPCLASS	Char	8	440	447	Class of the POE. Valid values are "TERMINAL", "CONSOLE", "JESINPUT", and "APPCPORT".
LINK_UTK_USER_ID	Char	8	449	456	User ID associated with the record.
LINK_UTK_GRP_ID	Char	8	458	465	Group ID associated with the record.
LINK_UTK_DFT_GRP	Yes/No	4	467	470	Is a default group assigned?
LINK_UTK_DFT_SECL	Yes/No	4	472	475	Is a default SECLABEL assigned?
LINK_APPC_LINK	Char	16	477	492	A key to link together audit records for a user's APPC transaction processing work.

Table 95. Format of the LINK Record Extension (continued)

Field Name	Type	Length	Position		Comments
			Start	End	
LINK_AUDIT_CODE	Char	11	494	504	Audit function code. For more information on the function codes, see Appendix I, "OpenExtensions Audit Function Codes," on page 439.
LINK_OLD_REAL_UID	Integer	10	506	515	Old real UID.
LINK_OLD_EFF_UID	Integer	10	517	526	Old effective UID.
LINK_OLD_SAVED_UID	Integer	10	528	537	Old saved UID.
LINK_OLD_REAL_GID	Integer	10	539	548	Old real GID.
LINK_OLD_EFF_GID	Integer	10	550	559	Old effective GID.
LINK_OLD_SAVED_GID	Integer	10	561	570	Old saved GID.
LINK_PATH_NAME	Char	1023	572	1594	The requested path name.
LINK_FILE_ID	Char	32	1596	1627	File ID.
LINK_FILE_OWN_UID	Integer	10	1629	1638	The owner UID associated with the file.
LINK_FILE_OWN_GID	Integer	10	1640	1649	The owner GID associated with the file.
LINK_REQUEST_PATH2	Char	1023	1651	2673	Second requested path name.
LINK_PATH_TYPE	Char	4	2675	2678	Type of the requested path name. Valid values are "OLD" and "NEW".
LINK_FILEPOOL	Char	8	2680	2687	SFS filepool containing the BFS file.
LINK_FILESPACE	Char	8	2689	2696	SFS filespace containing the BFS file.
LINK_INODE	Integer	10	2698	2707	Inode (file serial number).
LINK_SCID	Integer	10	2709	2718	File SCID.
LINK_DCE_LINK	Char	16	2720	2735	Link to connect DCE records that originate from a single DCE request.
LINK_AUTH_TYPE	Char	13	2737	2749	Defines the type of request. Valid values are: "SERVER" and "AUTH_CLIENT" and "UNAUTH_CLIENT".

Event Qualifiers for LINK Requests

The event qualifier that may be associated with a LINK event is shown in Table 96 on page 215.

Table 96. Event Code Qualifiers for LINK Records

Event Qualifier	Event Number	Event Description
SUCCESS	00	New link created. There are no failure cases for this event.

The Format of the MKDIR Record Extension

Table 97 on page 215 describes the format of a record that is created by making a directory.

Table 97. Format of the MKDIR Record Extension

Field Name	Type	Length	Position		Comments
			Start	End	
MDIR_CLASS	Char	8	282	289	Class name.
MDIR_USER_NAME	Char	20	291	310	The name associated with the user ID.
MDIR_UTK_ENCR	Yes/No	4	312	315	Is the UTOKEN associated with this user encrypted?

Table 97. Format of the MKDIR Record Extension (continued)

Field Name	Type	Length	Position		Comments
			Start	End	
MDIR_UTK_PRE19	Yes/No	4	317	320	Is this a token for a release earlier than RACF 1.9?
MDIR_UTK_VERPROF	Yes/No	4	322	325	Is the VERIFYX propagation flag set?
MDIR_UTK_NJEUNUSR	Yes/No	4	327	330	Is this the NJE undefined user?
MDIR_UTK_LOGUSR	Yes/No	4	332	335	Is UAUDIT specified for this user?
MDIR_UTK_SPECIAL	Yes/No	4	337	340	Is this a RACF SPECIAL user?
MDIR_UTK_DEFAULT	Yes/No	4	342	345	Is this a default token?
MDIR_UTK_UNKNUSR	Yes/No	4	347	350	Is this an undefined user?
MDIR_UTK_ERROR	Yes/No	4	352	355	Is this user token in error?
MDIR_UTK_TRUSTED	Yes/No	4	357	360	Is this user a part of the trusted computing base (TCB)?
MDIR_UTK_SESSTYPE	Char	8	362	369	The session type of this session.
MDIR_UTK_SURROGAT	Yes/No	4	371	374	Is this a surrogate user?
MDIR_UTK_REMOTE	Yes/No	4	376	379	Is this a remote job?
MDIR_UTK_PRIV	Yes/No	4	381	384	Is this a privileged user ID?
MDIR_UTK_SECL	Char	8	386	393	The SECLABEL of the user.
MDIR_UTK_EXECNODE	Char	8	395	402	The execution node of the work.
MDIR_UTK_SUSER_ID	Char	8	404	411	The submitting user ID.
MDIR_UTK_SNODE	Char	8	413	420	The submitting node.
MDIR_UTK_SGRP_ID	Char	8	422	429	The submitting group ID.
MDIR_UTK_SPOE	Char	8	431	438	The port of entry.
MDIR_UTK_SPCLASS	Char	8	440	447	Class of the POE. Valid values are "TERMINAL", "CONSOLE", "JESINPUT", and "APPCPORT".
MDIR_UTK_USER_ID	Char	8	449	456	User ID associated with the record.
MDIR_UTK_GRP_ID	Char	8	458	465	Group ID associated with the record.
MDIR_UTK_DFT_GRP	Yes/No	4	467	470	Is a default group assigned?
MDIR_UTK_DFT_SECL	Yes/No	4	472	475	Is a default SECLABEL assigned?
MDIR_APPC_LINK	Char	16	477	492	A key to link together audit records for a user's APPC transaction processing work.
MDIR_AUDIT_CODE	Char	11	494	504	Audit function code. For more information on the function codes, see Appendix I, "OpenExtensions Audit Function Codes," on page 439.
MDIR_OLD_REAL_UID	Integer	10	506	515	Old real UID.
MDIR_OLD_EFF_UID	Integer	10	517	526	Old effective UID.
MDIR_OLD_SAVED_UID	Integer	10	528	537	Old saved UID.
MDIR_OLD_REAL_GID	Integer	10	539	548	Old real GID.
MDIR_OLD_EFF_GID	Integer	10	550	559	Old effective GID.
MDIR_OLD_SAVED_GID	Integer	10	561	570	Old saved GID.
MDIR_PATH_NAME	Char	1023	572	1594	The requested path name.
MDIR_FILE_ID	Char	32	1596	1627	File ID.
MDIR_FILE_OWN_UID	Integer	10	1629	1638	The owner UID associated with the file.
MDIR_FILE_OWN_GID	Integer	10	1640	1649	The owner GID associated with the file.

Table 97. Format of the MKDIR Record Extension (continued)

Field Name	Type	Length	Position		Comments
			Start	End	
MDIR_OLD_S_ISGID	Yes/No	4	1651	1654	Was the S_ISGID bit requested on for this file?
MDIR_OLD_S_ISUID	Yes/No	4	1656	1659	Was the S_ISUID bit requested on for this file?
MDIR_OLD_S_ISVTX	Yes/No	4	1661	1664	Was the S_ISVTX bit requested on for this file?
MDIR_OLD_OWN_READ	Yes/No	4	1666	1669	Was the owner READ bit on for this file?
MDIR_OLD_OWN_WRITE	Yes/No	4	1671	1674	Was the owner WRITE bit on for this file?
MDIR_OLD_OWN_EXEC	Yes/No	4	1676	1679	Was the owner EXECUTE bit on for this file?
MDIR_OLD_GRP_READ	Yes/No	4	1681	1684	Was the group READ bit on for this file?
MDIR_OLD_GRP_WRITE	Yes/No	4	1686	1689	Was the group WRITE bit on for this file?
MDIR_OLD_GRP_EXEC	Yes/No	4	1691	1694	Was the group EXECUTE bit on for this file?
MDIR_OLD_OTH_READ	Yes/No	4	1696	1699	Was the other READ bit on for this file?
MDIR_OLD_OTH_WRITE	Yes/No	4	1701	1704	Was the other WRITE bit on for this file?
MDIR_OLD_OTH_EXEC	Yes/No	4	1706	1709	Was the other EXECUTE bit on for this file?
MDIR_NEW_S_ISGID	Yes/No	4	1711	1714	Is the S_ISGID bit requested on for this file?
MDIR_NEW_S_ISUID	Yes/No	4	1716	1719	Is the S_ISUID bit requested on for this file?
MDIR_NEW_S_ISVTX	Yes/No	4	1721	1724	Is the S_ISVTX bit requested on for this file?
MDIR_NEW_OWN_READ	Yes/No	4	1726	1729	Is the owner READ bit on for this file?
MDIR_NEW_OWN_WRITE	Yes/No	4	1731	1734	Is the owner WRITE bit on for this file?
MDIR_NEW_OWN_EXEC	Yes/No	4	1736	1739	Is the owner EXECUTE bit on for this file?
MDIR_NEW_GRP_READ	Yes/No	4	1741	1744	Is the group READ bit on for this file?
MDIR_NEW_GRP_WRITE	Yes/No	4	1746	1749	Is the group WRITE bit on for this file?
MDIR_NEW_GRP_EXEC	Yes/No	4	1751	1754	Is the group EXECUTE bit on for this file?
MDIR_NEW_OTH_READ	Yes/No	4	1756	1759	Is the other READ bit on for this file?
MDIR_NEW_OTH_WRITE	Yes/No	4	1761	1764	Is the other WRITE bit on for this file?
MDIR_NEW_OTH_EXEC	Yes/No	4	1766	1769	Is the other EXECUTE bit on for this file?
MDIR_UNEW_READ	Char	8	1771	1778	What are the new user audit options for READ actions? Valid values are "NONE", "SUCCESS", "FAIL", and "ALL".
MDIR_UNEW_WRITE	Char	8	1780	1787	What are the new user audit options for WRITE actions? Valid values are "NONE", "SUCCESS", "FAIL", and "ALL".
MDIR_UNEW_EXEC	Char	8	1789	1796	What are the new user audit options for EXECUTE actions? Valid values are "NONE", "SUCCESS", "FAIL", and "ALL".
MDIR_ANEW_READ	Char	8	1798	1805	What are the new auditor audit options for READ actions? Valid values are "NONE", "SUCCESS", "FAIL", and "ALL".
MDIR_ANEW_WRITE	Char	8	1807	1814	What are the new auditor audit options for WRITE actions? Valid values are "NONE", "SUCCESS", "FAIL", and "ALL".
MDIR_ANEW_EXEC	Char	8	1816	1823	What are the new auditor audit options for EXECUTE actions? Valid values are "NONE", "SUCCESS", "FAIL", and "ALL".
MDIR_REQ_S_ISGID	Yes/No	4	1825	1828	Was the S_ISGID bit requested on for this file?

Table 97. Format of the MKDIR Record Extension (continued)

Field Name	Type	Length	Position		Comments
			Start	End	
MDIR_REQ_S_ISUID	Yes/No	4	1830	1833	Was the S_ISUID bit requested on for this file?
MDIR_REQ_S_ISVTX	Yes/No	4	1835	1838	Was the S_ISVTX bit requested on for this file?
MDIR_REQ_OWN_READ	Yes/No	4	1840	1843	Was the owner READ bit requested on for this file?
MDIR_REQ_OWN_WRITE	Yes/No	4	1845	1848	Was the owner WRITE bit requested on for this file?
MDIR_REQ_OWN_EXEC	Yes/No	4	1850	1853	Was the owner EXECUTE bit requested on for this file?
MDIR_REQ_GRP_READ	Yes/No	4	1855	1858	Was the group READ bit requested on for this file?
MDIR_REQ_GRP_WRITE	Yes/No	4	1860	1863	Was the group WRITE bit requested on for this file?
MDIR_REQ_GRP_EXEC	Yes/No	4	1865	1868	Was the group EXECUTE bit requested on for this file?
MDIR_REQ_OTH_READ	Yes/No	4	1870	1873	Was the other READ bit requested on for this file?
MDIR_REQ_OTH_WRITE	Yes/No	4	1875	1878	Was the other WRITE bit requested on for this file?
MDIR_REQ_OTH_EXEC	Yes/No	4	1880	1883	Was the other EXECUTE bit requested on for this file?
MDIR_FILEPOOL	Char	8	1885	1892	SFS filepool containing the BFS file.
MDIR_FILESPACE	Char	8	1894	1901	SFS filespace containing the BFS file.
MDIR_INODE	Integer	10	1903	1912	Inode (file serial number).
MDIR_SCID	Integer	10	1914	1923	File SCID.

Event Qualifiers for MKDIR Requests

The event qualifier that may be associated with making a directory is shown in Table 98 on page 218.

Table 98. Event Code Qualifiers for MKDIR Records

Event Qualifier	Event Number	Event Description
SUCCESS	00	Directory created. There are no failure cases for this event.

The Format of the MKNOD Record Extension

Table 99 on page 218 describes the format of a record that is created by making a node.

Table 99. Format of the MKNOD Record Extension

Field Name	Type	Length	Position		Comments
			Start	End	
MNOD_CLASS	Char	8	282	289	Class name.
MNOD_USER_NAME	Char	20	291	310	The name associated with the user ID.
MNOD_UTK_ENCR	Yes/No	4	312	315	Is the UTOKEN associated with this user encrypted?
MNOD_UTK_PRE19	Yes/No	4	317	320	Is this a token for a release earlier than RACF 1.9?
MNOD_UTK_VERPROF	Yes/No	4	322	325	Is the VERIFYX propagation flag set?
MNOD_UTK_NJEUNUSR	Yes/No	4	327	330	Is this the NJE undefined user?
MNOD_UTK_LOGUSR	Yes/No	4	332	335	Is UAUDIT specified for this user?
MNOD_UTK_SPECIAL	Yes/No	4	337	340	Is this a RACF SPECIAL user?
MNOD_UTK_DEFAULT	Yes/No	4	342	345	Is this a default token?
MNOD_UTK_UNKNUSR	Yes/No	4	347	350	Is this an undefined user?

Table 99. Format of the MKNOD Record Extension (continued)

Field Name	Type	Length	Position		Comments
			Start	End	
MNOD_UTK_ERROR	Yes/No	4	352	355	Is this user token in error?
MNOD_UTK_TRUSTED	Yes/No	4	357	360	Is this user a part of the trusted computing base (TCB)?
MNOD_UTK_SESSTYPE	Char	8	362	369	The session type of this session.
MNOD_UTK_SURROGAT	Yes/No	4	371	374	Is this a surrogate user?
MNOD_UTK_REMOTE	Yes/No	4	376	379	Is this a remote job?
MNOD_UTK_PRIV	Yes/No	4	381	384	Is this a privileged user ID?
MNOD_UTK_SECL	Char	8	386	393	The SECLABEL of the user.
MNOD_UTK_EXECNODE	Char	8	395	402	The execution node of the work.
MNOD_UTK_SUSER_ID	Char	8	404	411	The submitting user ID.
MNOD_UTK_SNODE	Char	8	413	420	The submitting node.
MNOD_UTK_SGRP_ID	Char	8	422	429	The submitting group ID.
MNOD_UTK_SPOE	Char	8	431	438	The port of entry.
MNOD_UTK_SPCLASS	Char	8	440	447	Class of the POE. Valid values are "TERMINAL", "CONSOLE", "JESINPUT", and "APPCPORT".
MNOD_UTK_USER_ID	Char	8	449	456	User ID associated with the record.
MNOD_UTK_GRP_ID	Char	8	458	465	Group ID associated with the record.
MNOD_UTK_DFT_GRP	Yes/No	4	467	470	Is a default group assigned?
MNOD_UTK_DFT_SECL	Yes/No	4	472	475	Is a default SECLABEL assigned?
MNOD_APPC_LINK	Char	16	477	492	A key to link together audit records for a user's APPC transaction processing work.
MNOD_AUDIT_CODE	Char	11	494	504	Audit function code. For more information on the function codes, see Appendix I, “OpenExtensions Audit Function Codes,” on page 439.
MNOD_OLD_REAL_UID	Integer	10	506	515	Old real UID.
MNOD_OLD_EFF_UID	Integer	10	517	526	Old effective UID.
MNOD_OLD_SAVED_UID	Integer	10	528	537	Old saved UID.
MNOD_OLD_REAL_GID	Integer	10	539	548	Old real GID.
MNOD_OLD_EFF_GID	Integer	10	550	559	Old effective GID.
MNOD_OLD_SAVED_GID	Integer	10	561	570	Old saved GID.
MNOD_PATH_NAME	Char	1023	572	1594	The requested path name.
MNOD_FILE_ID	Char	32	1596	1627	File ID.
MNOD_FILE_OWN_UID	Integer	10	1629	1638	The owner UID associated with the file.
MNOD_FILE_OWN_GID	Integer	10	1640	1649	The owner GID associated with the file.
MNOD_OLD_S_ISGID	Yes/No	4	1651	1654	Was the S_ISGID bit requested on for this file?
MNOD_OLD_S_ISUID	Yes/No	4	1656	1659	Was the S_ISUID bit requested on for this file?
MNOD_OLD_S_ISVTX	Yes/No	4	1661	1664	Was the S_ISVTX bit requested on for this file?
MNOD_OLD_OWN_READ	Yes/No	4	1666	1669	Was the owner READ bit on for this file?
MNOD_OLD_OWN_WRITE	Yes/No	4	1671	1674	Was the owner WRITE bit on for this file?
MNOD_OLD_OWN_EXEC	Yes/No	4	1676	1679	Was the owner EXECUTE bit on for this file?
MNOD_OLD_GRP_READ	Yes/No	4	1681	1684	Was the group READ bit on for this file?

Table 99. Format of the MKNOD Record Extension (continued)

Field Name	Type	Length	Position		Comments
			Start	End	
MNOD_OLD_GRP_WRITE	Yes/No	4	1686	1689	Was the group WRITE bit on for this file?
MNOD_OLD_GRP_EXEC	Yes/No	4	1691	1694	Was the group EXECUTE bit on for this file?
MNOD_OLD_OTH_READ	Yes/No	4	1696	1699	Was the other READ bit on for this file?
MNOD_OLD_OTH_WRITE	Yes/No	4	1701	1704	Was the other WRITE bit on for this file?
MNOD_OLD_OTH_EXEC	Yes/No	4	1706	1709	Was the other EXECUTE bit on for this file?
MNOD_NEW_S_ISGID	Yes/No	4	1711	1714	Is the S_ISGID bit requested on for this file?
MNOD_NEW_S_ISUID	Yes/No	4	1716	1719	Is the S_ISUID bit requested on for this file?
MNOD_NEW_S_ISVTX	Yes/No	4	1721	1724	Is the S_ISVTX bit requested on for this file?
MNOD_NEW_OWN_READ	Yes/No	4	1726	1729	Is the owner READ bit on for this file?
MNOD_NEW_OWN_WRITE	Yes/No	4	1731	1734	Is the owner WRITE bit on for this file?
MNOD_NEW_OWN_EXEC	Yes/No	4	1736	1739	Is the owner EXECUTE bit on for this file?
MNOD_NEW_GRP_READ	Yes/No	4	1741	1744	Is the group READ bit on for this file?
MNOD_NEW_GRP_WRITE	Yes/No	4	1746	1749	Is the group WRITE bit on for this file?
MNOD_NEW_GRP_EXEC	Yes/No	4	1751	1754	Is the group EXECUTE bit on for this file?
MNOD_NEW_OTH_READ	Yes/No	4	1756	1759	Is the other READ bit on for this file?
MNOD_NEW_OTH_WRITE	Yes/No	4	1761	1764	Is the other WRITE bit on for this file?
MNOD_NEW_OTH_EXEC	Yes/No	4	1766	1769	Is the other EXECUTE bit on for this file?
MNOD_UNEW_READ	Char	8	1771	1778	What are the new user audit options for READ actions? Valid values are "NONE", "SUCCESS", "FAIL", and "ALL".
MNOD_UNEW_WRITE	Char	8	1780	1787	What are the new user audit options for WRITE actions? Valid values are "NONE", "SUCCESS", "FAIL", and "ALL".
MNOD_UNEW_EXEC	Char	8	1789	1796	What are the new user audit options for EXECUTE actions? Valid values are "NONE", "SUCCESS", "FAIL", and "ALL".
MNOD_ANEW_READ	Char	8	1798	1805	What are the new auditor audit options for READ actions? Valid values are "NONE", "SUCCESS", "FAIL", and "ALL".
MNOD_ANEW_WRITE	Char	8	1807	1814	What are the new auditor audit options for WRITE actions? Valid values are "NONE", "SUCCESS", "FAIL", and "ALL".
MNOD_ANEW_EXEC	Char	8	1816	1823	What are the new auditor audit options for EXECUTE actions? Valid values are "NONE", "SUCCESS", "FAIL", and "ALL".
MNOD_REQ_S_ISGID	Yes/No	4	1825	1828	Was the S_ISGID bit requested on for this file?
MNOD_REQ_S_ISUID	Yes/No	4	1830	1833	Was the S_ISUID bit requested on for this file?
MNOD_REQ_S_ISVTX	Yes/No	4	1835	1838	Was the S_ISVTX bit requested on for this file?
MNOD_REQ_OWN_READ	Yes/No	4	1840	1843	Was the owner READ bit requested on for this file?
MNOD_REQ_OWN_WRITE	Yes/No	4	1845	1848	Was the owner WRITE bit requested on for this file?
MNOD_REQ_OWN_EXEC	Yes/No	4	1850	1853	Was the owner EXECUTE bit requested on for this file?
MNOD_REQ_GRP_READ	Yes/No	4	1855	1858	Was the group READ bit requested on for this file?
MNOD_REQ_GRP_WRITE	Yes/No	4	1860	1863	Was the group WRITE bit requested on for this file?

Table 99. Format of the MKNOD Record Extension (continued)

Field Name	Type	Length	Position		Comments
			Start	End	
MNOD_REQ_GRP_EXEC	Yes/No	4	1865	1868	Was the group EXECUTE bit requested on for this file?
MNOD_REQ_OTH_READ	Yes/No	4	1870	1873	Was the other READ bit requested on for this file?
MNOD_REQ_OTH_WRITE	Yes/No	4	1875	1878	Was the other WRITE bit requested on for this file?
MNOD_REQ_OTH_EXEC	Yes/No	4	1880	1883	Was the other EXECUTE bit requested on for this file?
MNOD_FILEPOOL	Char	8	1885	1892	SFS filepool containing the BFS file.
MNOD_FILESPACE	Char	8	1894	1901	SFS filespace containing the BFS file.
MNOD_INODE	Integer	10	1903	1912	Inode (file serial number).
MNOD_SCID	Integer	10	1914	1923	File SCID.

Event Qualifiers for MKNOD Requests

The event qualifier that may be associated with making a node is shown in Table 100 on page 221.

Table 100. Event Code Qualifiers for MKNOD Records

Event Qualifier	Event Number	Event Description
SUCCESS	00	Node created. There are no failure cases for this event.

The Format of the Mount File System Record Extension

Table 101 on page 221 describes the format of a record that is created by mounting a file system.

Table 101. Format of the Mount File System Record Extension

Field Name	Type	Length	Position		Comments
			Start	End	
MFS_CLASS	Char	8	282	289	Class name.
MFS_USER_NAME	Char	20	291	310	The name associated with the user ID.
MFS_UTK_ENCR	Yes/No	4	312	315	Is the UTOKEN associated with this user encrypted?
MFS_UTK_PRE19	Yes/No	4	317	320	Is this a token for a release earlier than RACF 1.9?
MFS_UTK_VERPROF	Yes/No	4	322	325	Is the VERIFYX propagation flag set?
MFS_UTK_NJEUNUSR	Yes/No	4	327	330	Is this the NJE undefined user?
MFS_UTK_LOGUSR	Yes/No	4	332	335	Is UAUDIT specified for this user?
MFS_UTK_SPECIAL	Yes/No	4	337	340	Is this a RACF SPECIAL user?
MFS_UTK_DEFAULT	Yes/No	4	342	345	Is this a default token?
MFS_UTK_UNKNUSR	Yes/No	4	347	350	Is this an undefined user?
MFS_UTK_ERROR	Yes/No	4	352	355	Is this user token in error?
MFS_UTK_TRUSTED	Yes/No	4	357	360	Is this user a part of the trusted computing base (TCB)?
MFS_UTK_SESSTYPE	Char	8	362	369	The session type of this session.
MFS_UTK_SURROGAT	Yes/No	4	371	374	Is this a surrogate user?
MFS_UTK_REMOTE	Yes/No	4	376	379	Is this a remote job?
MFS_UTK_PRIV	Yes/No	4	381	384	Is this a privileged user ID?
MFS_UTK_SECL	Char	8	386	393	The SECLABEL of the user.

Table 101. Format of the Mount File System Record Extension (continued)

Field Name	Type	Length	Position		Comments
			Start	End	
MFS_UTK_EXECNODE	Char	8	395	402	The execution node of the work.
MFS_UTK_SUSER_ID	Char	8	404	411	The submitting user ID.
MFS_UTK_SNODE	Char	8	413	420	The submitting node.
MFS_UTK_SGRP_ID	Char	8	422	429	The submitting group ID.
MFS_UTK_SPOE	Char	8	431	438	The port of entry.
MFS_UTK_SPCLASS	Char	8	440	447	Class of the POE. Valid values are "TERMINAL", "CONSOLE", "JESINPUT", and "APPCPORT".
MFS_UTK_USER_ID	Char	8	449	456	User ID associated with the record.
MFS_UTK_GRP_ID	Char	8	458	465	Group ID associated with the record.
MFS_UTK_DFT_GRP	Yes/No	4	467	470	Is a default group assigned?
MFS_UTK_DFT_SECL	Yes/No	4	472	475	Is a default SECLABEL assigned?
MFS_APPC_LINK	Char	16	477	492	A key to link together audit records for a user's APPC transaction processing work.
MFS_AUDIT_CODE	Char	11	494	504	Audit function code. For more information on the function codes, see Appendix I, “OpenExtensions Audit Function Codes,” on page 439.
MFS_OLD_REAL_UID	Integer	10	506	515	Old real UID.
MFS_OLD_EFF_UID	Integer	10	517	526	Old effective UID.
MFS_OLD_SAVED_UID	Integer	10	528	537	Old saved UID.
MFS_OLD_REAL_GID	Integer	10	539	548	Old real GID.
MFS_OLD_EFF_GID	Integer	10	550	559	Old effective GID.
MFS_OLD_SAVED_GID	Integer	10	561	570	Old saved GID.
MFS_PATH_NAME	Char	1023	572	1594	The requested path name.
MFS_FILE_ID	Char	32	1596	1627	File ID.
MFS_FILE_OWN_UID	Integer	10	1629	1638	The owner UID associated with the file.
MFS_FILE_OWN_GID	Integer	10	1640	1649	The owner GID associated with the file.
MFS_HFS_DS_NAME	Char	44	1651	1694	HFS data set name for the mounted file system.
MFS_DCE_LINK	Char	16	1696	1711	Link to connect DCE records that originate from a single DCE request.
MFS_AUTH_TYPE	Char	13	1713	1725	Defines the type of request. Valid values are: "SERVER" and "AUTH_CLIENT" and "UNAUTH_CLIENT".

Event Qualifiers for Mount File System Requests

The event qualifier that may be associated with the mounting of a file system event is shown in Table 102 on page 222.

Table 102. Event Code Qualifiers for Mount File System Records

Event Qualifier	Event Number	Event Description
SUCCESS	00	File system mounted. There are no failure cases for this event.

The Format of the OPENFILE Record Extension

Table 103 on page 223 describes the format of a record that is created by opening a file.

Table 103. Format of the OPENFILE Extension

Field Name	Type	Length	Position		Comments
			Start	End	
OPEN_CLASS	Char	8	282	289	Class name.
OPEN_USER_NAME	Char	20	291	310	The name associated with the user ID.
OPEN_UTK_ENCR	Yes/No	4	312	315	Is the UTOKEN associated with this user encrypted?
OPEN_UTK_PRE19	Yes/No	4	317	320	Is this a token for a release earlier than RACF 1.9?
OPEN_UTK_VERPROF	Yes/No	4	322	325	Is the VERIFYX propagation flag set?
OPEN_UTK_NJEUNUSR	Yes/No	4	327	330	Is this the NJE undefined user?
OPEN_UTK_LOGUSR	Yes/No	4	332	335	Is UAUDIT specified for this user?
OPEN_UTK_SPECIAL	Yes/No	4	337	340	Is this a RACF SPECIAL user?
OPEN_UTK_DEFAULT	Yes/No	4	342	345	Is this a default token?
OPEN_UTK_UNKNUSR	Yes/No	4	347	350	Is this an undefined user?
OPEN_UTK_ERROR	Yes/No	4	352	355	Is this user token in error?
OPEN_UTK_TRUSTED	Yes/No	4	357	360	Is this user a part of the trusted computing base (TCB)?
OPEN_UTK_SESSTYPE	Char	8	362	369	The session type of this session.
OPEN_UTK_SURROGAT	Yes/No	4	371	374	Is this a surrogate user?
OPEN_UTK_REMOTE	Yes/No	4	376	379	Is this a remote job?
OPEN_UTK_PRIV	Yes/No	4	381	384	Is this a privileged user ID?
OPEN_UTK_SECL	Char	8	386	393	The SECLABEL of the user.
OPEN_UTK_EXECNODE	Char	8	395	402	The execution node of the work.
OPEN_UTK_SUSER_ID	Char	8	404	411	The submitting user ID.
OPEN_UTK_SNODE	Char	8	413	420	The submitting node.
OPEN_UTK_SGRP_ID	Char	8	422	429	The submitting group ID.
OPEN_UTK_SPOE	Char	8	431	438	The port of entry.
OPEN_UTK_SPCLASS	Char	8	440	447	Class of the POE. Valid values are "TERMINAL", "CONSOLE", "JESINPUT", and "APPCPORT".
OPEN_UTK_USER_ID	Char	8	449	456	User ID associated with the record.
OPEN_UTK_GRP_ID	Char	8	458	465	Group ID associated with the record.
OPEN_UTK_DFT_GRP	Yes/No	4	467	470	Is a default group assigned?
OPEN_UTK_DFT_SECL	Yes/No	4	472	475	Is a default SECLABEL assigned?
OPEN_APPC_LINK	Char	16	477	492	A key to link together audit records for a user's APPC transaction processing work.
OPEN_AUDIT_CODE	Char	11	494	504	Audit function code. For more information on the function codes, see Appendix I, “OpenExtensions Audit Function Codes,” on page 439.
OPEN_OLD_REAL_UID	Integer	10	506	515	Old real UID.
OPEN_OLD_EFF_UID	Integer	10	517	526	Old effective UID.
OPEN_OLD_SAVED_UID	Integer	10	528	537	Old saved UID.
OPEN_OLD_REAL_GID	Integer	10	539	548	Old real GID.

Table 103. Format of the OPENFILE Extension (continued)

Field Name	Type	Length	Position		Comments
			Start	End	
OPEN_OLD_EFF_GID	Integer	10	550	559	Old effective GID.
OPEN_OLD_SAVED_GID	Integer	10	561	570	Old saved GID.
OPEN_PATH_NAME	Char	1023	572	1594	The requested path name.
OPEN_FILE_ID	Char	32	1596	1627	File ID.
OPEN_FILE_OWN_UID	Integer	10	1629	1638	The owner UID associated with the file.
OPEN_FILE_OWN_GID	Integer	10	1640	1649	The owner GID associated with the file.
OPEN_OLD_S_ISGID	Yes/No	4	1651	1654	Was the S_ISGID bit requested on for this file?
OPEN_OLD_S_ISUID	Yes/No	4	1656	1659	Was the S_ISUID bit requested on for this file?
OPEN_OLD_S_ISVTX	Yes/No	4	1661	1664	Was the S_ISVTX bit requested on for this file?
OPEN_OLD_OWN_READ	Yes/No	4	1666	1669	Was the owner READ bit on for this file?
OPEN_OLD_OWN_WRITE	Yes/No	4	1671	1674	Was the owner WRITE bit on for this file?
OPEN_OLD_OWN_EXEC	Yes/No	4	1676	1679	Was the owner EXECUTE bit on for this file?
OPEN_OLD_GRP_READ	Yes/No	4	1681	1684	Was the group READ bit on for this file?
OPEN_OLD_GRP_WRITE	Yes/No	4	1686	1689	Was the group WRITE bit on for this file?
OPEN_OLD_GRP_EXEC	Yes/No	4	1691	1694	Was the group EXECUTE bit on for this file?
OPEN_OLD_OTH_READ	Yes/No	4	1696	1699	Was the other READ bit on for this file?
OPEN_OLD_OTH_WRITE	Yes/No	4	1701	1704	Was the other WRITE bit on for this file?
OPEN_OLD_OTH_EXEC	Yes/No	4	1706	1709	Was the other EXECUTE bit on for this file?
OPEN_NEW_S_ISGID	Yes/No	4	1711	1714	Is the S_ISGID bit requested on for this file?
OPEN_NEW_S_ISUID	Yes/No	4	1716	1719	Is the S_ISUID bit requested on for this file?
OPEN_NEW_S_ISVTX	Yes/No	4	1721	1724	Is the S_ISVTX bit requested on for this file?
OPEN_NEW_OWN_READ	Yes/No	4	1726	1729	Is the owner READ bit on for this file?
OPEN_NEW_OWN_WRITE	Yes/No	4	1731	1734	Is the owner WRITE bit on for this file?
OPEN_NEW_OWN_EXEC	Yes/No	4	1736	1739	Is the owner EXECUTE bit on for this file?
OPEN_NEW_GRP_READ	Yes/No	4	1741	1744	Is the group READ bit on for this file?
OPEN_NEW_GRP_WRITE	Yes/No	4	1746	1749	Is the group WRITE bit on for this file?
OPEN_NEW_GRP_EXEC	Yes/No	4	1751	1754	Is the group EXECUTE bit on for this file?
OPEN_NEW_OTH_READ	Yes/No	4	1756	1759	Is the other READ bit on for this file?
OPEN_NEW_OTH_WRITE	Yes/No	4	1761	1764	Is the other WRITE bit on for this file?
OPEN_NEW_OTH_EXEC	Yes/No	4	1766	1769	Is the other EXECUTE bit on for this file?
OPEN_UNEW_READ	Char	8	1771	1778	What are the new user audit options for READ actions? Valid values are "NONE", "SUCCESS", "FAIL", and "ALL".
OPEN_UNEW_WRITE	Char	8	1780	1787	What are the new user audit options for WRITE actions? Valid values are "NONE", "SUCCESS", "FAIL", and "ALL".
OPEN_UNEW_EXEC	Char	8	1789	1796	What are the new user audit options for EXECUTE actions? Valid values are "NONE", "SUCCESS", "FAIL", and "ALL".
OPEN_ANEW_READ	Char	8	1798	1805	What are the new auditor audit options for READ actions? Valid values are "NONE", "SUCCESS", "FAIL", and "ALL".

Table 103. Format of the OPENFILE Extension (continued)

Field Name	Type	Length	Position		Comments
			Start	End	
OPEN_ANEW_WRITE	Char	8	1807	1814	What are the new auditor audit options for WRITE actions? Valid values are "NONE", "SUCCESS", "FAIL", and "ALL".
OPEN_ANEW_EXEC	Char	8	1816	1823	What are the new auditor audit options for EXECUTE actions? Valid values are "NONE", "SUCCESS", "FAIL", and "ALL".
OPEN_REQ_S_ISGID	Yes/No	4	1825	1828	Was the S_ISGID bit requested on for this file?
OPEN_REQ_S_ISUID	Yes/No	4	1830	1833	Was the S_ISUID bit requested on for this file?
OPEN_REQ_S_ISVTX	Yes/No	4	1835	1838	Was the S_ISVTX bit requested on for this file?
OPEN_REQ_OWN_READ	Yes/No	4	1840	1843	Was the owner READ bit requested on for this file?
OPEN_REQ_OWN_WRITE	Yes/No	4	1845	1848	Was the owner WRITE bit requested on for this file?
OPEN_REQ_OWN_EXEC	Yes/No	4	1850	1853	Was the owner EXECUTE bit requested on for this file?
OPEN_REQ_GRP_READ	Yes/No	4	1855	1858	Was the group READ bit requested on for this file?
OPEN_REQ_GRP_WRITE	Yes/No	4	1860	1863	Was the group WRITE bit requested on for this file?
OPEN_REQ_GRP_EXEC	Yes/No	4	1865	1868	Was the group EXECUTE bit requested on for this file?
OPEN_REQ_OTH_READ	Yes/No	4	1870	1873	Was the other READ bit requested on for this file?
OPEN_REQ_OTH_WRITE	Yes/No	4	1875	1878	Was the other WRITE bit requested on for this file?
OPEN_REQ_OTH_EXEC	Yes/No	4	1880	1883	Was the other EXECUTE bit requested on for this file?
OPEN_FILEPOOL	Char	8	1885	1892	SFS filepool containing the BFS file.
OPEN_FILESPACE	Char	8	1894	1901	SFS filespace containing the BFS file.
OPEN_INODE	Integer	10	1903	1912	Inode (file serial number).
OPEN_SCID	Integer	10	1914	1923	File SCID.

Event Qualifiers for OPENFILE Requests

The event qualifier that may be associated with making a node is shown in Table 104 on page 225.

Table 104. Event Code Qualifiers for OPENFILE Records

Event Qualifier	Event Number	Event Description
SUCCESS	00	File created. There are no failure cases for this event.

The Format of the PTRACE Record Extension

Table 105 on page 225 describes the format of a record that is created by the tracing of a process.

Table 105. Format of the PTRACE Record Extension

Field Name	Type	Length	Position		Comments
			Start	End	
PTRC_CLASS	Char	8	282	289	Class name.
PTRC_USER_NAME	Char	20	291	310	The name associated with the user ID.
PTRC_UTK_ENCR	Yes/No	4	312	315	Is the UTOKEN associated with this user encrypted?
PTRC_UTK_PRE19	Yes/No	4	317	320	Is this a token for a release earlier than RACF 1.9?
PTRC_UTK_VERPROF	Yes/No	4	322	325	Is the VERIFYX propagation flag set?

Table 105. Format of the PTRACE Record Extension (continued)

Field Name	Type	Length	Position		Comments
			Start	End	
PTRC_UTK_NJEUNUSR	Yes/No	4	327	330	Is this the NJE undefined user?
PTRC_UTK_LOGUSR	Yes/No	4	332	335	Is UAUDIT specified for this user?
PTRC_UTK_SPECIAL	Yes/No	4	337	340	Is this a RACF SPECIAL user?
PTRC_UTK_DEFAULT	Yes/No	4	342	345	Is this a default token?
PTRC_UTK_UNKNUSR	Yes/No	4	347	350	Is this an undefined user?
PTRC_UTK_ERROR	Yes/No	4	352	355	Is this user token in error?
PTRC_UTK_TRUSTED	Yes/No	4	357	360	Is this user a part of the trusted computing base (TCB)?
PTRC_UTK_SESSTYPE	Char	8	362	369	The session type of this session.
PTRC_UTK_SURROGAT	Yes/No	4	371	374	Is this a surrogate user?
PTRC_UTK_REMOTE	Yes/No	4	376	379	Is this a remote job?
PTRC_UTK_PRIV	Yes/No	4	381	384	Is this a privileged user ID?
PTRC_UTK_SECL	Char	8	386	393	The SECLABEL of the user.
PTRC_UTK_EXECNODE	Char	8	395	402	The execution node of the work.
PTRC_UTK_SUSER_ID	Char	8	404	411	The submitting user ID.
PTRC_UTK_SNODE	Char	8	413	420	The submitting node.
PTRC_UTK_SGRP_ID	Char	8	422	429	The submitting group ID.
PTRC_UTK_SPOE	Char	8	431	438	The port of entry.
PTRC_UTK_SPCLASS	Char	8	440	447	Class of the POE. Valid values are "TERMINAL", "CONSOLE", "JESINPUT", and "APPCPORT".
PTRC_UTK_USER_ID	Char	8	449	456	User ID associated with the record.
PTRC_UTK_GRP_ID	Char	8	458	465	Group ID associated with the record.
PTRC_UTK_DFT_GRP	Yes/No	4	467	470	Is a default group assigned?
PTRC_UTK_DFT_SECL	Yes/No	4	472	475	Is a default SECLABEL assigned?
PTRC_APPC_LINK	Char	16	477	492	Key to link together APPC records.
PTRC_AUDIT_CODE	Char	11	494	504	Audit function code. For more information on the function codes, see Appendix I, “OpenExtensions Audit Function Codes,” on page 439.
PTRC_OLD_REAL_UID	Integer	10	506	515	Old real UID.
PTRC_OLD_EFF_UID	Integer	10	517	526	Old effective UID.
PTRC_OLD_SAVED_UID	Integer	10	528	537	Old saved UID.
PTRC_OLD_REAL_GID	Integer	10	539	548	Old real GID.
PTRC_OLD_EFF_GID	Integer	10	550	559	Old effective GID.
PTRC_OLD_SAVED_GID	Integer	10	561	570	Old saved GID.
PTRC_TGT_REAL_UID	Integer	10	572	581	Target real UID.
PTRC_TGT_EFF_UID	Integer	10	583	592	Target effective UID.
PTRC_TGT_SAVED_UID	Integer	10	594	603	Target saved UID.
PTRC_TGT_REAL_GID	Integer	10	605	614	Target real GID.
PTRC_TGT_EFF_GID	Integer	10	616	625	Target effective GID.
PTRC_TGT_SAVED_GID	Integer	10	627	636	Target saved GID.

Table 105. Format of the PTRACE Record Extension (continued)					
Field Name	**Type**	**Length**	**Position**		**Comments**
			Start	**End**	
PTRC_TGT_PID	Integer	10	638	647	Target process ID.

Event Qualifiers for the PTRACE Process Record Extension

The event qualifiers that may be associated with the tracing of a process are shown in Table 106 on page 227.

Table 106. Event Code Qualifiers for PTRACE Records		
Event Qualifier	**Event Number**	**Event Description**
SUCCESS	00	Access allowed.
NOTAUTH	01	Not authorized to trace the specified process.

The Format of the Rename File Record Extension

Table 95 on page 214 describes the format of a record that is created by a rename operation.

Table 107. Format of the Rename File Record Extension					
Field Name	**Type**	**Length**	**Position**		**Comments**
			Start	**End**	
RENF_CLASS	Char	8	282	289	Class name.
RENF_USER_NAME	Char	20	291	310	The name associated with the user ID.
RENF_UTK_ENCR	Yes/No	4	312	315	Is the UTOKEN associated with this user encrypted?
RENF_UTK_PRE19	Yes/No	4	317	320	Is this a token for a release earlier than RACF 1.9?
RENF_UTK_VERPROF	Yes/No	4	322	325	Is the VERIFYX propagation flag set?
RENF_UTK_NJEUNUSR	Yes/No	4	327	330	Is this the NJE undefined user?
RENF_UTK_LOGUSR	Yes/No	4	332	335	Is UAUDIT specified for this user?
RENF_UTK_SPECIAL	Yes/No	4	337	340	Is this a RACF SPECIAL user?
RENF_UTK_DEFAULT	Yes/No	4	342	345	Is this a default token?
RENF_UTK_UNKNUSR	Yes/No	4	347	350	Is this an undefined user?
RENF_UTK_ERROR	Yes/No	4	352	355	Is this user token in error?
RENF_UTK_TRUSTED	Yes/No	4	357	360	Is this user a part of the trusted computing base (TCB)?
RENF_UTK_SESSTYPE	Char	8	362	369	The session type of this session.
RENF_UTK_SURROGAT	Yes/No	4	371	374	Is this a surrogate user?
RENF_UTK_REMOTE	Yes/No	4	376	379	Is this a remote job?
RENF_UTK_PRIV	Yes/No	4	381	384	Is this a privileged user ID?
RENF_UTK_SECL	Char	8	386	393	The SECLABEL of the user.
RENF_UTK_EXECNODE	Char	8	395	402	The execution node of the work.
RENF_UTK_SUSER_ID	Char	8	404	411	The submitting user ID.
RENF_UTK_SNODE	Char	8	413	420	The submitting node.
RENF_UTK_SGRP_ID	Char	8	422	429	The submitting group ID.
RENF_UTK_SPOE	Char	8	431	438	The port of entry.

Table 107. Format of the Rename File Record Extension (continued)

Field Name	Type	Length	Position		Comments
			Start	End	
RENF_UTK_SPCLASS	Char	8	440	447	Class of the POE. Valid values are "TERMINAL", "CONSOLE", "JESINPUT", and "APPCPORT".
RENF_UTK_USER_ID	Char	8	449	456	User ID associated with the record.
RENF_UTK_GRP_ID	Char	8	458	465	Group ID associated with the record.
RENF_UTK_DFT_GRP	Yes/No	4	467	470	Is a default group assigned?
RENF_UTK_DFT_SECL	Yes/No	4	472	475	Is a default SECLABEL assigned?
RENF_APPC_LINK	Char	16	477	492	A key to link together audit records for a user's APPC transaction processing work.
RENF_AUDIT_CODE	Char	11	494	504	Audit function code. For more information on the function codes, see Appendix I, "OpenExtensions Audit Function Codes," on page 439.
RENF_OLD_REAL_UID	Integer	10	506	515	Old real UID.
RENF_OLD_EFF_UID	Integer	10	517	526	Old effective UID.
RENF_OLD_SAVED_UID	Integer	10	528	537	Old saved UID.
RENF_OLD_REAL_GID	Integer	10	539	548	Old real GID.
RENF_OLD_EFF_GID	Integer	10	550	559	Old effective GID.
RENF_OLD_SAVED_GID	Integer	10	561	570	Old saved GID.
RENF_PATH_NAME	Char	1023	572	1594	The requested path name.
RENF_FILE_ID	Char	32	1596	1627	File ID.
RENF_FILE_OWN_UID	Integer	10	1629	1638	The owner UID associated with the file.
RENF_FILE_OWN_GID	Integer	10	1640	1649	The owner GID associated with the file.
RENF_PATH2	Char	1023	1651	2673	Second requested path name.
RENF_FILE_ID2	Char	32	2675	2706	Second requested file ID.
RENF_OWNER_UID	Integer	10	2708	2717	UID of the owner of the deleted file.
RENF_OWNER_GID	Integer	10	2719	2728	GID of the owner of the deleted file.
RENF_PATH_TYPE	Char	4	2730	2733	Type of the requested path name. Valid values are "OLD" and "NEW".
RENF_LAST_DELETED	Yes/No	4	2735	2738	Was the last link deleted?
RENF_FILEPOOL	Char	8	2740	2747	SFS filepool containing the BFS file.
RENF_FILESPACE	Char	8	2749	2756	SFS filespace containing the BFS file.
RENF_INODE	Integer	10	2758	2767	Inode (file serial number).
RENF_SCID	Integer	10	2769	2778	File SCID.
RENF_FILEPOOL2	Char	8	2780	2787	SFS filepool containing the second BFS file.
RENF_FILESPACE2	Char	8	2789	2796	SFS filespace containing the second BFS file.
RENF_INODE2	Integer	10	2798	2807	Second Inode (file serial number).
RENF_SCID2	Integer	10	2809	2818	Second File SCID.
RENF_DCE_LINK	Char	16	2820	2835	Link to connect DCE records that originate from a single DCE request.
RENF_AUTH_TYPE	Char	13	2837	2849	Defines the type of request. Valid values are: "SERVER" and "AUTH_CLIENT" and "UNAUTH_CLIENT".

Event Qualifiers for Rename File Requests

The event qualifier that may be associated with a file rename event is shown in Table 108 on page 229.

Table 108. Event Code Qualifiers for Rename File Records

Event Qualifier	Event Number	Event Description
SUCCESS	00	File renamed. There are no failure cases for this event.

The Format of the RMDIR Record Extension

Table 109 on page 229 describes the format of a record that is created by removing a directory.

Table 109. Format of the RMDIR Record Extension

Field Name	Type	Length	Position		Comments
			Start	End	
RDIR_CLASS	Char	8	282	289	Class name.
RDIR_USER_NAME	Char	20	291	310	The name associated with the user ID.
RDIR_UTK_ENCR	Yes/No	4	312	315	Is the UTOKEN associated with this user encrypted?
RDIR_UTK_PRE19	Yes/No	4	317	320	Is this a token for a release earlier than RACF 1.9?
RDIR_UTK_VERPROF	Yes/No	4	322	325	Is the VERIFYX propagation flag set?
RDIR_UTK_NJEUNUSR	Yes/No	4	327	330	Is this the NJE undefined user?
RDIR_UTK_LOGUSR	Yes/No	4	332	335	Is UAUDIT specified for this user?
RDIR_UTK_SPECIAL	Yes/No	4	337	340	Is this a RACF SPECIAL user?
RDIR_UTK_DEFAULT	Yes/No	4	342	345	Is this a default token?
RDIR_UTK_UNKNUSR	Yes/No	4	347	350	Is this an undefined user?
RDIR_UTK_ERROR	Yes/No	4	352	355	Is this user token in error?
RDIR_UTK_TRUSTED	Yes/No	4	357	360	Is this user a part of the trusted computing base (TCB)?
RDIR_UTK_SESSTYPE	Char	8	362	369	The session type of this session.
RDIR_UTK_SURROGAT	Yes/No	4	371	374	Is this a surrogate user?
RDIR_UTK_REMOTE	Yes/No	4	376	379	Is this a remote job?
RDIR_UTK_PRIV	Yes/No	4	381	384	Is this a privileged user ID?
RDIR_UTK_SECL	Char	8	386	393	The SECLABEL of the user.
RDIR_UTK_EXECNODE	Char	8	395	402	The execution node of the work.
RDIR_UTK_SUSER_ID	Char	8	404	411	The submitting user ID.
RDIR_UTK_SNODE	Char	8	413	420	The submitting node.
RDIR_UTK_SGRP_ID	Char	8	422	429	The submitting group ID.
RDIR_UTK_SPOE	Char	8	431	438	The port of entry.
RDIR_UTK_SPCLASS	Char	8	440	447	Class of the POE. Valid values are "TERMINAL", "CONSOLE", "JESINPUT", and "APPCPORT".
RDIR_UTK_USER_ID	Char	8	449	456	User ID associated with the record.
RDIR_UTK_GRP_ID	Char	8	458	465	Group ID associated with the record.
RDIR_UTK_DFT_GRP	Yes/No	4	467	470	Is a default group assigned?
RDIR_UTK_DFT_SECL	Yes/No	4	472	475	Is a default SECLABEL assigned?

Table 109. Format of the RMDIR Record Extension (continued)

Field Name	Type	Length	Position		Comments
			Start	End	
RDIR_APPC_LINK	Char	16	477	492	A key to link together audit records for a user's APPC transaction processing work.
RDIR_AUDIT_CODE	Char	11	494	504	Audit function code. For more information on the function codes, see Appendix I, "OpenExtensions Audit Function Codes," on page 439.
RDIR_OLD_REAL_UID	Integer	10	506	515	Old real UID.
RDIR_OLD_EFF_UID	Integer	10	517	526	Old effective UID.
RDIR_OLD_SAVED_UID	Integer	10	528	537	Old saved UID.
RDIR_OLD_REAL_GID	Integer	10	539	548	Old real GID.
RDIR_OLD_EFF_GID	Integer	10	550	559	Old effective GID.
RDIR_OLD_SAVED_GID	Integer	10	561	570	Old saved GID.
RDIR_PATH_NAME	Char	1023	572	1594	The requested path name.
RDIR_FILE_ID	Char	32	1596	1627	File ID.
RDIR_FILE_OWN_UID	Integer	10	1629	1638	The owner UID associated with the file.
RDIR_FILE_OWN_GID	Integer	10	1640	1649	The owner GID associated with the file.
RDIR_FILEPOOL	Char	8	1651	1658	SFS filepool containing the BFS file.
RDIR_FILESPACE	Char	8	1660	1667	SFS filespace containing the BFS file.
RDIR_INODE	Integer	10	1669	1678	Inode (file serial number).
RDIR_SCID	Integer	10	1680	1689	File SCID.
RDIR_DCE_LINK	Char	16	1691	1706	Link to connect DCE records that originate from a single DCE request.
RDIR_AUTH_TYPE	Char	13	1708	1720	Defines the type of request. Valid values are: "SERVER" and "AUTH_CLIENT" and "UNAUTH_CLIENT".

Event Qualifiers for RMDIR Requests

The event qualifier that may be associated with removing a directory is shown in Table 110 on page 230.

Table 110. Event Code Qualifiers for RMDIR Records

Event Qualifier	Event Number	Event Description
SUCCESS	00	Directory removed. There are no failure cases for this event.

The Format of the SETEGID Record Extension

Table 111 on page 230 describes the format of a record that is created by the setting of an effective GID.

Table 111. Format of the SETEGID Record Extension

Field Name	Type	Length	Position		Comments
			Start	End	
SEGI_CLASS	Char	8	282	289	Class name.
SEGI_USER_NAME	Char	20	291	310	The name associated with the user ID.
SEGI_UTK_ENCR	Yes/No	4	312	315	Is the UTOKEN associated with this user encrypted?
SEGI_UTK_PRE19	Yes/No	4	317	320	Is this a token for a release earlier than RACF 1.9?

Table 111. Format of the SETEGID Record Extension (continued)

Field Name	Type	Length	Position		Comments
			Start	End	
SEGI_UTK_VERPROF	Yes/No	4	322	325	Is the VERIFYX propagation flag set?
SEGI_UTK_NJEUNUSR	Yes/No	4	327	330	Is this the NJE undefined user?
SEGI_UTK_LOGUSR	Yes/No	4	332	335	Is UAUDIT specified for this user?
SEGI_UTK_SPECIAL	Yes/No	4	337	340	Is this a RACF SPECIAL user?
SEGI_UTK_DEFAULT	Yes/No	4	342	345	Is this a default token?
SEGI_UTK_UNKNUSR	Yes/No	4	347	350	Is this an undefined user?
SEGI_UTK_ERROR	Yes/No	4	352	355	Is this user token in error?
SEGI_UTK_TRUSTED	Yes/No	4	357	360	Is this user a part of the trusted computing base (TCB)?
SEGI_UTK_SESSTYPE	Char	8	362	369	The session type of this session.
SEGI_UTK_SURROGAT	Yes/No	4	371	374	Is this a surrogate user?
SEGI_UTK_REMOTE	Yes/No	4	376	379	Is this a remote job?
SEGI_UTK_PRIV	Yes/No	4	381	384	Is this a privileged user ID?
SEGI_UTK_SECL	Char	8	386	393	The SECLABEL of the user.
SEGI_UTK_EXECNODE	Char	8	395	402	The execution node of the work.
SEGI_UTK_SUSER_ID	Char	8	404	411	The submitting user ID.
SEGI_UTK_SNODE	Char	8	413	420	The submitting node.
SEGI_UTK_SGRP_ID	Char	8	422	429	The submitting group ID.
SEGI_UTK_SPOE	Char	8	431	438	The port of entry.
SEGI_UTK_SPCLASS	Char	8	440	447	Class of the POE. Valid values are "TERMINAL", "CONSOLE", "JESINPUT", and "APPCPORT".
SEGI_UTK_USER_ID	Char	8	449	456	User ID associated with the record.
SEGI_UTK_GRP_ID	Char	8	458	465	Group ID associated with the record.
SEGI_UTK_DFT_GRP	Yes/No	4	467	470	Is a default group assigned?
SEGI_UTK_DFT_SECL	Yes/No	4	472	475	Is a default SECLABEL assigned?
SEGI_APPC_LINK	Char	16	477	492	Key to link together APPC records.
SEGI_AUDIT_CODE	Char	11	494	504	Audit function code. For more information on the function codes, see Appendix I, “OpenExtensions Audit Function Codes,” on page 439.
SEGI_OLD_REAL_UID	Integer	10	506	515	Old real UID.
SEGI_OLD_EFF_UID	Integer	10	517	526	Old effective UID.
SEGI_OLD_SAVED_UID	Integer	10	528	537	Old saved UID.
SEGI_OLD_REAL_GID	Integer	10	539	548	Old real GID.
SEGI_OLD_EFF_GID	Integer	10	550	559	Old effective GID.
SEGI_OLD_SAVED_GID	Integer	10	561	570	Old saved GID.
SEGI_NEW_REAL_GID	Integer	10	572	581	New real GID.
SEGI_NEW_EFF_GID	Integer	10	583	592	New effective GID.
SEGI_NEW_SAVED_GID	Integer	10	594	603	New saved GID.
SEGI_GID	Integer	10	605	614	The GID input parameter.

Event Qualifiers for the SETEGID Record Extension

The event qualifiers that may be associated with setting the effective GID are shown in Table 112 on page 232.

Table 112. Event Code Qualifiers for SETEGID Records

Event Qualifier	Event Number	Event Description
SUCCESS	00	Successful change of effective GID.
NOTAUTH	01	Not authorized to set the effective GID.

The Format of the SETEUID Record Extension

Table 113 on page 232 describes the format of a record that is created by the setting of an effective UID.

Table 113. Format of the SETEUID Record Extension

Field Name	Type	Length	Position		Comments
			Start	End	
SEUI_CLASS	Char	8	282	289	Class name.
SEUI_USER_NAME	Char	20	291	310	The name associated with the user ID.
SEUI_UTK_ENCR	Yes/No	4	312	315	Is the UTOKEN associated with this user encrypted?
SEUI_UTK_PRE19	Yes/No	4	317	320	Is this a token for a release earlier than RACF 1.9?
SEUI_UTK_VERPROF	Yes/No	4	322	325	Is the VERIFYX propagation flag set?
SEUI_UTK_NJEUNUSR	Yes/No	4	327	330	Is this the NJE undefined user?
SEUI_UTK_LOGUSR	Yes/No	4	332	335	Is UAUDIT specified for this user?
SEUI_UTK_SPECIAL	Yes/No	4	337	340	Is this a RACF SPECIAL user?
SEUI_UTK_DEFAULT	Yes/No	4	342	345	Is this a default token?
SEUI_UTK_UNKNUSR	Yes/No	4	347	350	Is this an undefined user?
SEUI_UTK_ERROR	Yes/No	4	352	355	Is this user token in error?
SEUI_UTK_TRUSTED	Yes/No	4	357	360	Is this user a part of the trusted computing base (TCB)?
SEUI_UTK_SESSTYPE	Char	8	362	369	The session type of this session.
SEUI_UTK_SURROGAT	Yes/No	4	371	374	Is this a surrogate user?
SEUI_UTK_REMOTE	Yes/No	4	376	379	Is this a remote job?
SEUI_UTK_PRIV	Yes/No	4	381	384	Is this a privileged user ID?
SEUI_UTK_SECL	Char	8	386	393	The SECLABEL of the user.
SEUI_UTK_EXECNODE	Char	8	395	402	The execution node of the work.
SEUI_UTK_SUSER_ID	Char	8	404	411	The submitting user ID.
SEUI_UTK_SNODE	Char	8	413	420	The submitting node.
SEUI_UTK_SGRP_ID	Char	8	422	429	The submitting group ID.
SEUI_UTK_SPOE	Char	8	431	438	The port of entry.
SEUI_UTK_SPCLASS	Char	8	440	447	Class of the POE. Valid values are "TERMINAL", "CONSOLE", "JESINPUT", and "APPCPORT".
SEUI_UTK_USER_ID	Char	8	449	456	User ID associated with the record.
SEUI_UTK_GRP_ID	Char	8	458	465	Group ID associated with the record.
SEUI_UTK_DFT_GRP	Yes/No	4	467	470	Is a default group assigned?

Table 113. Format of the SETEUID Record Extension (continued)

Field Name	Type	Length	Position		Comments
			Start	End	
SEUI_UTK_DFT_SECL	Yes/No	4	472	475	Is a default SECLABEL assigned?
SEUI_APPC_LINK	Char	16	477	492	Key to link together APPC records.
SEUI_AUDIT_CODE	Char	11	494	504	Audit function code. For more information on the function codes, see Appendix I, "OpenExtensions Audit Function Codes," on page 439.
SEUI_OLD_REAL_UID	Integer	10	506	515	Old real UID.
SEUI_OLD_EFF_UID	Integer	10	517	526	Old effective UID.
SEUI_OLD_SAVED_UID	Integer	10	528	537	Old saved UID.
SEUI_OLD_REAL_GID	Integer	10	539	548	Old real GID.
SEUI_OLD_EFF_GID	Integer	10	550	559	Old effective GID.
SEUI_OLD_SAVED_GID	Integer	10	561	570	Old saved GID.
SEUI_NEW_REAL_UID	Integer	10	572	581	New real UID.
SEUI_NEW_EFF_UID	Integer	10	583	592	New effective UID.
SEUI_NEW_SAVED_UID	Integer	10	594	603	New saved UID.
SEUI_UID	Integer	10	605	614	The UID input parameter.

Event Qualifiers for the SETEUID Record Extension

The event qualifiers that may be associated with setting the effective UID are shown in Table 114 on page 233.

Table 114. Event Code Qualifiers for SETEUID Records

Event Qualifier	Event Number	Event Description
SUCCESS	00	Successful change of UIDs.
NOTAUTH	01	Not authorized to set the effective UID.

The Format of the SETGID Record Extension

Table 115 on page 233 describes the format of a record that is created by the setting of a GID.

Table 115. Format of the SETGID Record Extension

Field Name	Type	Length	Position		Comments
			Start	End	
SGI_CLASS	Char	8	282	289	Class name.
SGI_USER_NAME	Char	20	291	310	The name associated with the user ID.
SGI_UTK_ENCR	Yes/No	4	312	315	Is the UTOKEN associated with this user encrypted?
SGI_UTK_PRE19	Yes/No	4	317	320	Is this a token for a release earlier than RACF 1.9?
SGI_UTK_VERPROF	Yes/No	4	322	325	Is the VERIFYX propagation flag set?
SGI_UTK_NJEUNUSR	Yes/No	4	327	330	Is this the NJE undefined user?
SGI_UTK_LOGUSR	Yes/No	4	332	335	Is UAUDIT specified for this user?
SGI_UTK_SPECIAL	Yes/No	4	337	340	Is this a RACF SPECIAL user?
SGI_UTK_DEFAULT	Yes/No	4	342	345	Is this a default token?
SGI_UTK_UNKNUSR	Yes/No	4	347	350	Is this an undefined user?

Table 115. Format of the SETGID Record Extension (continued)

Field Name	Type	Length	Position		Comments
			Start	End	
SGI_UTK_ERROR	Yes/No	4	352	355	Is this user token in error?
SGI_UTK_TRUSTED	Yes/No	4	357	360	Is this user a part of the trusted computing base (TCB)?
SGI_UTK_SESSTYPE	Char	8	362	369	The session type of this session.
SGI_UTK_SURROGAT	Yes/No	4	371	374	Is this a surrogate user?
SGI_UTK_REMOTE	Yes/No	4	376	379	Is this a remote job?
SGI_UTK_PRIV	Yes/No	4	381	384	Is this a privileged user ID?
SGI_UTK_SECL	Char	8	386	393	The SECLABEL of the user.
SGI_UTK_EXECNODE	Char	8	395	402	The execution node of the work.
SGI_UTK_SUSER_ID	Char	8	404	411	The submitting user ID.
SGI_UTK_SNODE	Char	8	413	420	The submitting node.
SGI_UTK_SGRP_ID	Char	8	422	429	The submitting group ID.
SGI_UTK_SPOE	Char	8	431	438	The port of entry.
SGI_UTK_SPCLASS	Char	8	440	447	Class of the POE. Valid values are "TERMINAL", "CONSOLE", "JESINPUT", and "APPCPORT".
SGI_UTK_USER_ID	Char	8	449	456	User ID associated with the record.
SGI_UTK_GRP_ID	Char	8	458	465	Group ID associated with the record.
SGI_UTK_DFT_GRP	Yes/No	4	467	470	Is a default group assigned?
SGI_UTK_DFT_SECL	Yes/No	4	472	475	Is a default SECLABEL assigned?
SGI_APPC_LINK	Char	16	477	492	Key to link together APPC records.
SGI_AUDIT_CODE	Char	11	494	504	Audit function code. For more information on the function codes, see Appendix I, “OpenExtensions Audit Function Codes,” on page 439.
SGI_OLD_REAL_UID	Integer	10	506	515	Old real UID.
SGI_OLD_EFF_UID	Integer	10	517	526	Old effective UID.
SGI_OLD_SAVED_UID	Integer	10	528	537	Old saved UID.
SGI_OLD_REAL_GID	Integer	10	539	548	Old real GID.
SGI_OLD_EFF_GID	Integer	10	550	559	Old effective GID.
SGI_OLD_SAVED_GID	Integer	10	561	570	Old saved GID.
SGI_NEW_REAL_GID	Integer	10	572	581	New real GID.
SGI_NEW_EFF_GID	Integer	10	583	592	New effective GID.
SGI_NEW_SAVED_GID	Integer	10	594	603	New saved GID.
SGI_GID	Integer	10	605	614	The GID input parameter.

Event Qualifiers for the SETGID Record Extension

The event qualifiers that may be associated with setting the GID are shown in Table 116 on page 234.

Table 116. Event Code Qualifiers for SETGID Records

Event Qualifier	Event Number	Event Description
SUCCESS	00	Successful change of GID.

Table 116. Event Code Qualifiers for SETGID Records (continued)		
Event Qualifier	**Event Number**	**Event Description**
NOTAUTH	01	Not authorized to set the GID.

The Format of the SETUID Record Extension

Table 117 on page 235 describes the format of a record that is created by the setting of a UID.

Table 117. Format of the SETUID Record Extension

Field Name	Type	Length	Position		Comments
			Start	End	
SUI_CLASS	Char	8	282	289	Class name.
SUI_USER_NAME	Char	20	291	310	The name associated with the user ID.
SUI_UTK_ENCR	Yes/No	4	312	315	Is the UTOKEN associated with this user encrypted?
SUI_UTK_PRE19	Yes/No	4	317	320	Is this a token for a release earlier than RACF 1.9?
SUI_UTK_VERPROF	Yes/No	4	322	325	Is the VERIFYX propagation flag set?
SUI_UTK_NJEUNUSR	Yes/No	4	327	330	Is this the NJE undefined user?
SUI_UTK_LOGUSR	Yes/No	4	332	335	Is UAUDIT specified for this user?
SUI_UTK_SPECIAL	Yes/No	4	337	340	Is this a RACF SPECIAL user?
SUI_UTK_DEFAULT	Yes/No	4	342	345	Is this a default token?
SUI_UTK_UNKNUSR	Yes/No	4	347	350	Is this an undefined user?
SUI_UTK_ERROR	Yes/No	4	352	355	Is this user token in error?
SUI_UTK_TRUSTED	Yes/No	4	357	360	Is this user a part of the trusted computing base (TCB)?
SUI_UTK_SESSTYPE	Char	8	362	369	The session type of this session.
SUI_UTK_SURROGAT	Yes/No	4	371	374	Is this a surrogate user?
SUI_UTK_REMOTE	Yes/No	4	376	379	Is this a remote job?
SUI_UTK_PRIV	Yes/No	4	381	384	Is this a privileged user ID?
SUI_UTK_SECL	Char	8	386	393	The SECLABEL of the user.
SUI_UTK_EXECNODE	Char	8	395	402	The execution node of the work.
SUI_UTK_SUSER_ID	Char	8	404	411	The submitting user ID.
SUI_UTK_SNODE	Char	8	413	420	The submitting node.
SUI_UTK_SGRP_ID	Char	8	422	429	The submitting group ID.
SUI_UTK_SPOE	Char	8	431	438	The port of entry.
SUI_UTK_SPCLASS	Char	8	440	447	Class of the POE. Valid values are "TERMINAL", "CONSOLE", "JESINPUT", and "APPCPORT".
SUI_UTK_USER_ID	Char	8	449	456	User ID associated with the record.
SUI_UTK_GRP_ID	Char	8	458	465	Group ID associated with the record.
SUI_UTK_DFT_GRP	Yes/No	4	467	470	Is a default group assigned?
SUI_UTK_DFT_SECL	Yes/No	4	472	475	Is a default SECLABEL assigned?
SUI_APPC_LINK	Char	16	477	492	Key to link together APPC records.
SUI_AUDIT_CODE	Char	11	494	504	Audit function code. For more information on the function codes, see Appendix I, “OpenExtensions Audit Function Codes,” on page 439.

Table 117. Format of the SETUID Record Extension (continued)

Field Name	Type	Length	Position		Comments
			Start	End	
SUI_OLD_REAL_UID	Integer	10	506	515	Old real UID.
SUI_OLD_EFF_UID	Integer	10	517	526	Old effective UID.
SUI_OLD_SAVED_UID	Integer	10	528	537	Old saved UID.
SUI_OLD_REAL_GID	Integer	10	539	548	Old real GID.
SUI_OLD_EFF_GID	Integer	10	550	559	Old effective GID.
SUI_OLD_SAVED_GID	Integer	10	561	570	Old saved GID.
SUI_NEW_REAL_UID	Integer	10	572	581	New real UID.
SUI_NEW_EFF_UID	Integer	10	583	592	New effective UID.
SUI_NEW_SAVED_UID	Integer	10	594	603	New saved UID.
SUI_UID	Integer	10	605	614	The UID input parameter.

Event Qualifiers for the SETUID Record Extension

The event qualifiers that may be associated with setting the effective UID are shown in Table 118 on page 236.

Table 118. Event Code Qualifiers for SETUID Records

Event Qualifier	Event Number	Event Description
SUCCESS	00	Successful change of UID.
NOTAUTH	01	Not authorized to set the UID.

The Format of the SYMLINK Record Extension

Table 119 on page 236 describes the format of a record that is created by a SYMLINK operation.

Table 119. Format of the SYMLINK Record Extension

Field Name	Type	Length	Position		Comments
			Start	End	
SYML_CLASS	Char	8	282	289	Class name.
SYML_USER_NAME	Char	20	291	310	The name associated with the user ID.
SYML_UTK_ENCR	Yes/No	4	312	315	Is the UTOKEN associated with this user encrypted?
SYML_UTK_PRE19	Yes/No	4	317	320	Is this a token for a release earlier than RACF 1.9?
SYML_UTK_VERPROF	Yes/No	4	322	325	Is the VERIFYX propagation flag set?
SYML_UTK_NJEUNUSR	Yes/No	4	327	330	Is this the NJE undefined user?
SYML_UTK_LOGUSR	Yes/No	4	332	335	Is UAUDIT specified for this user?
SYML_UTK_SPECIAL	Yes/No	4	337	340	Is this a RACF SPECIAL user?
SYML_UTK_DEFAULT	Yes/No	4	342	345	Is this a default token?
SYML_UTK_UNKNUSR	Yes/No	4	347	350	Is this an undefined user?
SYML_UTK_ERROR	Yes/No	4	352	355	Is this user token in error?
SYML_UTK_TRUSTED	Yes/No	4	357	360	Is this user a part of the trusted computing base (TCB)?
SYML_UTK_SESSTYPE	Char	8	362	369	The session type of this session.

Table 119. Format of the SYMLINK Record Extension (continued)

Field Name	Type	Length	Position		Comments
			Start	End	
SYML_UTK_SURROGAT	Yes/No	4	371	374	Is this a surrogate user?
SYML_UTK_REMOTE	Yes/No	4	376	379	Is this a remote job?
SYML_UTK_PRIV	Yes/No	4	381	384	Is this a privileged user ID?
SYML_UTK_SECL	Char	8	386	393	The SECLABEL of the user.
SYML_UTK_EXECNODE	Char	8	395	402	The execution node of the work.
SYML_UTK_SUSER_ID	Char	8	404	411	The submitting user ID.
SYML_UTK_SNODE	Char	8	413	420	The submitting node.
SYML_UTK_SGRP_ID	Char	8	422	429	The submitting group ID.
SYML_UTK_SPOE	Char	8	431	438	The port of entry.
SYML_UTK_SPCLASS	Char	8	440	447	Class of the POE. Valid values are "TERMINAL", "CONSOLE", "JESINPUT", and "APPCPORT".
SYML_UTK_USER_ID	Char	8	449	456	User ID associated with the record.
SYML_UTK_GRP_ID	Char	8	458	465	Group ID associated with the record.
SYML_UTK_DFT_GRP	Yes/No	4	467	470	Is a default group assigned?
SYML_UTK_DFT_SECL	Yes/No	4	472	475	Is a default SECLABEL assigned?
SYML_APPC_LINK	Char	16	477	492	A key to link together audit records for a user's APPC transaction processing work.
SYML_AUDIT_CODE	Char	11	494	504	Audit function code. For more information on the function codes, see Appendix I, "OpenExtensions Audit Function Codes," on page 439.
SYML_OLD_REAL_UID	Integer	10	506	515	Old real UID.
SYML_OLD_EFF_UID	Integer	10	517	526	Old effective UID.
SYML_OLD_SAVED_UID	Integer	10	528	537	Old saved UID.
SYML_OLD_REAL_GID	Integer	10	539	548	Old real GID.
SYML_OLD_EFF_GID	Integer	10	550	559	Old effective GID.
SYML_OLD_SAVED_GID	Integer	10	561	570	Old saved GID.
SYML_PATH_NAME	Char	1023	572	1594	The requested path name.
SYML_FILE_ID	Char	32	1596	1627	File ID.
SYML_FILE_OWN_UID	Integer	10	1629	1638	The owner UID associated with the file.
SYML_FILE_OWN_GID	Integer	10	1640	1649	The owner GID associated with the file.
SYML_SYMLINK_DATA	Char	1023	1651	2673	Content of SYMLINK.
SYML_FILEPOOL	Char	8	2675	2682	SFS filepool containing the BFS file.
SYML_FILESPACE	Char	8	2684	2691	SFS filespace containing the BFS file.
SYML_INODE	Integer	10	2693	2702	Inode (file serial number).
SYML_SCID	Integer	10	2704	2713	File SCID.
SYML_DCE_LINK	Char	16	2715	2730	Link to connect DCE records that originate from a single DCE request.
SYML_AUTH_TYPE	Char	13	2732	2744	Defines the type of request. Valid values are: "SERVER" and "AUTH_CLIENT" and "UNAUTH_CLIENT".

Event Qualifiers for SYMLINK Requests

The event qualifier that may be associated with a SYMLINK event is shown in Table 120 on page 238.

Table 120. Event Code Qualifiers for SYMLINK Records

Event Qualifier	Event Number	Event Description
SUCCESS	00	Successful SYMLINK. There are no failure cases for this event.

The Format of the UNLINK Record Extension

Table 121 on page 238 describes the format of a record that is created by an UNLINK operation.

Table 121. Format of the UNLINK Record Extension

Field Name	Type	Length	Position		Comments
			Start	End	
UNL_CLASS	Char	8	282	289	Class name.
UNL_USER_NAME	Char	20	291	310	The name associated with the user ID.
UNL_UTK_ENCR	Yes/No	4	312	315	Is the UTOKEN associated with this user encrypted?
UNL_UTK_PRE19	Yes/No	4	317	320	Is this a token for a release earlier than RACF 1.9?
UNL_UTK_VERPROF	Yes/No	4	322	325	Is the VERIFYX propagation flag set?
UNL_UTK_NJEUNUSR	Yes/No	4	327	330	Is this the NJE undefined user?
UNL_UTK_LOGUSR	Yes/No	4	332	335	Is UAUDIT specified for this user?
UNL_UTK_SPECIAL	Yes/No	4	337	340	Is this a RACF SPECIAL user?
UNL_UTK_DEFAULT	Yes/No	4	342	345	Is this a default token?
UNL_UTK_UNKNUSR	Yes/No	4	347	350	Is this an undefined user?
UNL_UTK_ERROR	Yes/No	4	352	355	Is this user token in error?
UNL_UTK_TRUSTED	Yes/No	4	357	360	Is this user a part of the trusted computing base (TCB)?
UNL_UTK_SESSTYPE	Char	8	362	369	The session type of this session.
UNL_UTK_SURROGAT	Yes/No	4	371	374	Is this a surrogate user?
UNL_UTK_REMOTE	Yes/No	4	376	379	Is this a remote job?
UNL_UTK_PRIV	Yes/No	4	381	384	Is this a privileged user ID?
UNL_UTK_SECL	Char	8	386	393	The SECLABEL of the user.
UNL_UTK_EXECNODE	Char	8	395	402	The execution node of the work.
UNL_UTK_SUSER_ID	Char	8	404	411	The submitting user ID.
UNL_UTK_SNODE	Char	8	413	420	The submitting node.
UNL_UTK_SGRP_ID	Char	8	422	429	The submitting group ID.
UNL_UTK_SPOE	Char	8	431	438	The port of entry.
UNL_UTK_SPCLASS	Char	8	440	447	Class of the POE. Valid values are "TERMINAL", "CONSOLE", "JESINPUT", and "APPCPORT".
UNL_UTK_USER_ID	Char	8	449	456	User ID associated with the record.
UNL_UTK_GRP_ID	Char	8	458	465	Group ID associated with the record.
UNL_UTK_DFT_GRP	Yes/No	4	467	470	Is a default group assigned?
UNL_UTK_DFT_SECL	Yes/No	4	472	475	Is a default SECLABEL assigned?

Table 121. Format of the UNLINK Record Extension (continued)

Field Name	Type	Length	Position		Comments
			Start	End	
UNL_APPC_LINK	Char	16	477	492	A key to link together audit records for a user's APPC transaction processing work.
UNL_AUDIT_CODE	Char	11	494	504	Audit function code. For more information on the function codes, see Appendix I, "OpenExtensions Audit Function Codes," on page 439.
UNL_OLD_REAL_UID	Integer	10	506	515	Old real UID.
UNL_OLD_EFF_UID	Integer	10	517	526	Old effective UID.
UNL_OLD_SAVED_UID	Integer	10	528	537	Old saved UID.
UNL_OLD_REAL_GID	Integer	10	539	548	Old real GID.
UNL_OLD_EFF_GID	Integer	10	550	559	Old effective GID.
UNL_OLD_SAVED_GID	Integer	10	561	570	Old saved GID.
UNL_PATH_NAME	Char	1023	572	1594	The requested path name.
UNL_FILE_ID	Char	32	1596	1627	File ID.
UNL_FILE_OWN_UID	Integer	10	1629	1638	The owner UID associated with the file.
UNL_FILE_OWN_GID	Integer	10	1640	1649	The owner GID associated with the file.
UNL_LAST_DELETED	Yes/No	4	1651	1654	Was the last link deleted?
UNL_FILEPOOL	Char	8	1656	1663	SFS filepool containing the BFS file.
UNL_FILESPACE	Char	8	1665	1672	SFS filespace containing the BFS file.
UNL_INODE	Integer	10	1674	1683	Inode (file serial number).
UNL_SCID	Integer	10	1685	1694	File SCID.
UNL_DCE_LINK	Char	16	1696	1711	Link to connect DCE records that originate from a single DCE request.
UNL_AUTH_TYPE	Char	13	1713	1725	Defines the type of request. Valid values are: "SERVER" and "AUTH_CLIENT" and "UNAUTH_CLIENT".

Event Qualifiers for UNLINK Requests

The event qualifier that may be associated with an UNLINK event is shown in Table 122 on page 239.

Table 122. Event Code Qualifiers for UNLINK Records

Event Qualifier	Event Number	Event Description
SUCCESS	00	Successful UNLINK. Failures are logged as check access event types.

The Format of the Unmount File System Record Extension

Table 123 on page 239 describes the format of a record that is created unmounting a file system.

Table 123. Format of the Unmount File System Record Extension

Field Name	Type	Length	Position		Comments
			Start	End	
UFS_CLASS	Char	8	282	289	Class name.
UFS_USER_NAME	Char	20	291	310	The name associated with the user ID.
UFS_UTK_ENCR	Yes/No	4	312	315	Is the UTOKEN associated with this user encrypted?

Table 123. Format of the Unmount File System Record Extension (continued)

Field Name	Type	Length	Position		Comments
			Start	End	
UFS_UTK_PRE19	Yes/No	4	317	320	Is this a token for a release earlier than RACF 1.9?
UFS_UTK_VERPROF	Yes/No	4	322	325	Is the VERIFYX propagation flag set?
UFS_UTK_NJEUNUSR	Yes/No	4	327	330	Is this the NJE undefined user?
UFS_UTK_LOGUSR	Yes/No	4	332	335	Is UAUDIT specified for this user?
UFS_UTK_SPECIAL	Yes/No	4	337	340	Is this a RACF SPECIAL user?
UFS_UTK_DEFAULT	Yes/No	4	342	345	Is this a default token?
UFS_UTK_UNKNUSR	Yes/No	4	347	350	Is this an undefined user?
UFS_UTK_ERROR	Yes/No	4	352	355	Is this user token in error?
UFS_UTK_TRUSTED	Yes/No	4	357	360	Is this user a part of the trusted computing base (TCB)?
UFS_UTK_SESSTYPE	Char	8	362	369	The session type of this session.
UFS_UTK_SURROGAT	Yes/No	4	371	374	Is this a surrogate user?
UFS_UTK_REMOTE	Yes/No	4	376	379	Is this a remote job?
UFS_UTK_PRIV	Yes/No	4	381	384	Is this a privileged user ID?
UFS_UTK_SECL	Char	8	386	393	The SECLABEL of the user.
UFS_UTK_EXECNODE	Char	8	395	402	The execution node of the work.
UFS_UTK_SUSER_ID	Char	8	404	411	The submitting user ID.
UFS_UTK_SNODE	Char	8	413	420	The submitting node.
UFS_UTK_SGRP_ID	Char	8	422	429	The submitting group ID.
UFS_UTK_SPOE	Char	8	431	438	The port of entry.
UFS_UTK_SPCLASS	Char	8	440	447	Class of the POE. Valid values are "TERMINAL", "CONSOLE", "JESINPUT", and "APPCPORT".
UFS_UTK_USER_ID	Char	8	449	456	User ID associated with the record.
UFS_UTK_GRP_ID	Char	8	458	465	Group ID associated with the record.
UFS_UTK_DFT_GRP	Yes/No	4	467	470	Is a default group assigned?
UFS_UTK_DFT_SECL	Yes/No	4	472	475	Is a default SECLABEL assigned?
UFS_APPC_LINK	Char	16	477	492	A key to link together audit records for a user's APPC transaction processing work.
UFS_AUDIT_CODE	Char	11	494	504	Audit function code. For more information on the function codes, see Appendix I, "OpenExtensions Audit Function Codes," on page 439.
UFS_OLD_REAL_UID	Integer	10	506	515	Old real UID.
UFS_OLD_EFF_UID	Integer	10	517	526	Old effective UID.
UFS_OLD_SAVED_UID	Integer	10	528	537	Old saved UID.
UFS_OLD_REAL_GID	Integer	10	539	548	Old real GID.
UFS_OLD_EFF_GID	Integer	10	550	559	Old effective GID.
UFS_OLD_SAVED_GID	Integer	10	561	570	Old saved GID.
UFS_PATH_NAME	Char	1023	572	1594	The requested path name.
UFS_FILE_ID	Char	32	1596	1627	File ID.
UFS_FILE_OWN_UID	Integer	10	1629	1638	The owner UID associated with the file.
UFS_FILE_OWN_GID	Integer	10	1640	1649	The owner GID associated with the file.

Table 123. Format of the Unmount File System Record Extension (continued)

Field Name	Type	Length	Position		Comments
			Start	End	
UFS_HFS_DS_NAME	Char	44	1651	1694	HFS data set name for the mounted file system.
UFS_DCE_LINK	Char	16	1696	1711	Link to connect DCE records that originate from a single DCE request.
UFS_AUTH_TYPE	Char	13	1713	1725	Defines the type of request. Valid values are: "SERVER" and "AUTH_CLIENT" and "UNAUTH_CLIENT".

Event Qualifiers for Unmount File System Requests

The event qualifier that may be associated with the unmounting of a file system is shown in Table 124 on page 241.

Table 124. Event Code Qualifiers for Unmount File System Records

Event Qualifier	Event Number	Event Description
SUCCESS	00	Unmount successful. Failures are logged as CKPRIV events.

The Format of the Check File Owner Record Extension

Table 125 on page 241 describes the format of a record that is created by checking the owner of a file.

Table 125. Format of the Check File Owner Record Extension

Field Name	Type	Length	Position		Comments
			Start	End	
CFOW_CLASS	Char	8	282	289	Class name.
CFOW_USER_NAME	Char	20	291	310	The name associated with the user ID.
CFOW_UTK_ENCR	Yes/No	4	312	315	Is the UTOKEN associated with this user encrypted?
CFOW_UTK_PRE19	Yes/No	4	317	320	Is this a token for a release earlier than RACF 1.9?
CFOW_UTK_VERPROF	Yes/No	4	322	325	Is the VERIFYX propagation flag set?
CFOW_UTK_NJEUNUSR	Yes/No	4	327	330	Is this the NJE undefined user?
CFOW_UTK_LOGUSR	Yes/No	4	332	335	Is UAUDIT specified for this user?
CFOW_UTK_SPECIAL	Yes/No	4	337	340	Is this a RACF SPECIAL user?
CFOW_UTK_DEFAULT	Yes/No	4	342	345	Is this a default token?
CFOW_UTK_UNKNUSR	Yes/No	4	347	350	Is this an undefined user?
CFOW_UTK_ERROR	Yes/No	4	352	355	Is this user token in error?
CFOW_UTK_TRUSTED	Yes/No	4	357	360	Is this user a part of the trusted computing base (TCB)?
CFOW_UTK_SESSTYPE	Char	8	362	369	The session type of this session.
CFOW_UTK_SURROGAT	Yes/No	4	371	374	Is this a surrogate user?
CFOW_UTK_REMOTE	Yes/No	4	376	379	Is this a remote job?
CFOW_UTK_PRIV	Yes/No	4	381	384	Is this a privileged user ID?
CFOW_UTK_SECL	Char	8	386	393	The SECLABEL of the user.
CFOW_UTK_EXECNODE	Char	8	395	402	The execution node of the work.
CFOW_UTK_SUSER_ID	Char	8	404	411	The submitting user ID.

Table 125. Format of the Check File Owner Record Extension (continued)

Field Name	Type	Length	Position		Comments
			Start	End	
CFOW_UTK_SNODE	Char	8	413	420	The submitting node.
CFOW_UTK_SGRP_ID	Char	8	422	429	The submitting group ID.
CFOW_UTK_SPOE	Char	8	431	438	The port of entry.
CFOW_UTK_SPCLASS	Char	8	440	447	Class of the POE. Valid values are "TERMINAL", "CONSOLE", "JESINPUT", and "APPCPORT".
CFOW_UTK_USER_ID	Char	8	449	456	User ID associated with the record.
CFOW_UTK_GRP_ID	Char	8	458	465	Group ID associated with the record.
CFOW_UTK_DFT_GRP	Yes/No	4	467	470	Is a default group assigned?
CFOW_UTK_DFT_SECL	Yes/No	4	472	475	Is a default SECLABEL assigned?
CFOW_APPC_LINK	Char	16	477	492	Key to link together APPC records.
CFOW_AUDIT_CODE	Char	11	494	504	Audit function code. For more information on the function codes, see Appendix I, “OpenExtensions Audit Function Codes,” on page 439.
CFOW_OLD_REAL_UID	Integer	10	506	515	Old real UID.
CFOW_OLD_EFF_UID	Integer	10	517	526	Old effective UID.
CFOW_OLD_SAVED_UID	Integer	10	528	537	Old saved UID.
CFOW_OLD_REAL_GID	Integer	10	539	548	Old real GID.
CFOW_OLD_EFF_GID	Integer	10	550	559	Old effective GID.
CFOW_OLD_SAVED_GID	Integer	10	561	570	Old saved GID.
CFOW_PATH_NAME	Char	1023	572	1594	The requested path name.
CFOW_FILE_ID	Char	32	1596	1627	File ID.
CFOW_FILE_OWN_UID	Integer	10	1629	1638	The owner UID associated with the file.
CFOW_FILE_OWN_GID	Integer	10	1640	1649	The owner GID associated with the file.
CFOW_FILEPOOL	Char	8	1651	1658	SFS filepool containing the BFS file.
CFOW_FILESPACE	Char	8	1660	1667	SFS filespace containing the BFS file.
CFOW_INODE	Integer	10	1669	1678	Inode (file serial number).
CFOW_SCID	Integer	10	1680	1689	File SCID.
CFOW_DCE_LINK	Char	16	1691	1706	Link to connect DCE records that originate from a single DCE request.
CFOW_AUTH_TYPE	Char	13	1708	1720	Defines the type of request. Valid values are: "SERVER" and "AUTH_CLIENT" and "UNAUTH_CLIENT".

Event Qualifiers for Check File Owner Requests

The event qualifiers that may be associated with checking a file's owner are shown in Table 126 on page 242.

Table 126. Event Code Qualifiers for Check File Owner Records

Event Qualifier	Event Number	Event Description
OWNER	00	The user is the owner.
NOTOWNER	01	The user is not the owner.

The Format of the Check Privilege Record Extension

Table 127 on page 243 describes the format of a record that is created by checking a user's privileges.

Table 127. Format of the Check Privilege Record Extension

Field Name	Type	Length	Position		Comments
			Start	End	
CPRV_CLASS	Char	8	282	289	Class name.
CPRV_USER_NAME	Char	20	291	310	The name associated with the user ID.
CPRV_UTK_ENCR	Yes/No	4	312	315	Is the UTOKEN associated with this user encrypted?
CPRV_UTK_PRE19	Yes/No	4	317	320	Is this a token for a release earlier than RACF 1.9?
CPRV_UTK_VERPROF	Yes/No	4	322	325	Is the VERIFYX propagation flag set?
CPRV_UTK_NJEUNUSR	Yes/No	4	327	330	Is this the NJE undefined user?
CPRV_UTK_LOGUSR	Yes/No	4	332	335	Is UAUDIT specified for this user?
CPRV_UTK_SPECIAL	Yes/No	4	337	340	Is this a RACF SPECIAL user?
CPRV_UTK_DEFAULT	Yes/No	4	342	345	Is this a default token?
CPRV_UTK_UNKNUSR	Yes/No	4	347	350	Is this an undefined user?
CPRV_UTK_ERROR	Yes/No	4	352	355	Is this user token in error?
CPRV_UTK_TRUSTED	Yes/No	4	357	360	Is this user a part of the trusted computing base (TCB)?
CPRV_UTK_SESSTYPE	Char	8	362	369	The session type of this session.
CPRV_UTK_SURROGAT	Yes/No	4	371	374	Is this a surrogate user?
CPRV_UTK_REMOTE	Yes/No	4	376	379	Is this a remote job?
CPRV_UTK_PRIV	Yes/No	4	381	384	Is this a privileged user ID?
CPRV_UTK_SECL	Char	8	386	393	The SECLABEL of the user.
CPRV_UTK_EXECNODE	Char	8	395	402	The execution node of the work.
CPRV_UTK_SUSER_ID	Char	8	404	411	The submitting user ID.
CPRV_UTK_SNODE	Char	8	413	420	The submitting node.
CPRV_UTK_SGRP_ID	Char	8	422	429	The submitting group ID.
CPRV_UTK_SPOE	Char	8	431	438	The port of entry.
CPRV_UTK_SPCLASS	Char	8	440	447	Class of the POE. Valid values are "TERMINAL", "CONSOLE", "JESINPUT", and "APPCPORT".
CPRV_UTK_USER_ID	Char	8	449	456	User ID associated with the record.
CPRV_UTK_GRP_ID	Char	8	458	465	Group ID associated with the record.
CPRV_UTK_DFT_GRP	Yes/No	4	467	470	Is a default group assigned?
CPRV_UTK_DFT_SECL	Yes/No	4	472	475	Is a default SECLABEL assigned?
CPRV_APPC_LINK	Char	16	477	492	A key to link together audit records for a user's APPC transaction processing work.
CPRV_AUDIT_CODE	Char	11	494	504	Audit function code. For more information on the function codes, see Appendix I, "OpenExtensions Audit Function Codes," on page 439.
CPRV_OLD_REAL_UID	Integer	10	506	515	Old real UID.
CPRV_OLD_EFF_UID	Integer	10	517	526	Old effective UID.
CPRV_OLD_SAVED_UID	Integer	10	528	537	Old saved UID.
CPRV_OLD_REAL_GID	Integer	10	539	548	Old real GID.

Table 127. Format of the Check Privilege Record Extension (continued)					
Field Name	**Type**	**Length**	**Position**		**Comments**
			Start	**End**	
CPRV_OLD_EFF_GID	Integer	10	550	559	Old effective GID.
CPRV_OLD_SAVED_GID	Integer	10	561	570	Old saved GID.
CPRV_DCE_LINK	Char	16	572	587	Link to connect DCE records that originate from a single DCE request.
CPRV_AUTH_TYPE	Char	13	589	601	Defines the type of request. Valid values are: "SERVER" and "AUTH_CLIENT" and "UNAUTH_CLIENT".

Event Qualifiers for Check Privilege Requests

The event qualifiers that may be associated with checking a user's privileges are shown in Table 128 on page 244.

Table 128. Event Code Qualifiers for Check Privilege Records		
Event Qualifier	**Event Number**	**Event Description**
SUCCESS	00	User is authorized.
NOTAUTH	01	The user is not authorized to the function.

The Format of the Open Slave TTY Record Extension

Table 129 on page 244 describes the format of a record that is created by the opening of a slave TTY.

Table 129. Format of the Open Slave TTY Record Extension					
Field Name	**Type**	**Length**	**Position**		**Comments**
			Start	**End**	
OSTY_CLASS	Char	8	282	289	Class name.
OSTY_USER_NAME	Char	20	291	310	The name associated with the user ID.
OSTY_UTK_ENCR	Yes/No	4	312	315	Is the UTOKEN associated with this user encrypted?
OSTY_UTK_PRE19	Yes/No	4	317	320	Is this a token for a release earlier than RACF 1.9?
OSTY_UTK_VERPROF	Yes/No	4	322	325	Is the VERIFYX propagation flag set?
OSTY_UTK_NJEUNUSR	Yes/No	4	327	330	Is this the NJE undefined user?
OSTY_UTK_LOGUSR	Yes/No	4	332	335	Is UAUDIT specified for this user?
OSTY_UTK_SPECIAL	Yes/No	4	337	340	Is this a RACF SPECIAL user?
OSTY_UTK_DEFAULT	Yes/No	4	342	345	Is this a default token?
OSTY_UTK_UNKNUSR	Yes/No	4	347	350	Is this an undefined user?
OSTY_UTK_ERROR	Yes/No	4	352	355	Is this user token in error?
OSTY_UTK_TRUSTED	Yes/No	4	357	360	Is this user a part of the trusted computing base (TCB)?
OSTY_UTK_SESSTYPE	Char	8	362	369	The session type of this session.
OSTY_UTK_SURROGAT	Yes/No	4	371	374	Is this a surrogate user?
OSTY_UTK_REMOTE	Yes/No	4	376	379	Is this a remote job?
OSTY_UTK_PRIV	Yes/No	4	381	384	Is this a privileged user ID?
OSTY_UTK_SECL	Char	8	386	393	The SECLABEL of the user.

Table 129. Format of the Open Slave TTY Record Extension (continued)

Field Name	Type	Length	Position		Comments
			Start	End	
OSTY_UTK_EXECNODE	Char	8	395	402	The execution node of the work.
OSTY_UTK_SUSER_ID	Char	8	404	411	The submitting user ID.
OSTY_UTK_SNODE	Char	8	413	420	The submitting node.
OSTY_UTK_SGRP_ID	Char	8	422	429	The submitting group ID.
OSTY_UTK_SPOE	Char	8	431	438	The port of entry.
OSTY_UTK_SPCLASS	Char	8	440	447	Class of the POE. Valid values are "TERMINAL", "CONSOLE", "JESINPUT", and "APPCPORT".
OSTY_UTK_USER_ID	Char	8	449	456	User ID associated with the record.
OSTY_UTK_GRP_ID	Char	8	458	465	Group ID associated with the record.
OSTY_UTK_DFT_GRP	Yes/No	4	467	470	Is a default group assigned?
OSTY_UTK_DFT_SECL	Yes/No	4	472	475	Is a default SECLABEL assigned?
OSTY_APPC_LINK	Char	16	477	492	A key to link together audit records for a user's APPC transaction processing work.
OSTY_AUDIT_CODE	Char	11	494	504	Audit function code. For more information on the function codes, see Appendix I, “OpenExtensions Audit Function Codes,” on page 439.
OSTY_OLD_REAL_UID	Integer	10	506	515	Old real UID.
OSTY_OLD_EFF_UID	Integer	10	517	526	Old effective UID.
OSTY_OLD_SAVED_UID	Integer	10	528	537	Old saved UID.
OSTY_OLD_REAL_GID	Integer	10	539	548	Old real GID.
OSTY_OLD_EFF_GID	Integer	10	550	559	Old effective GID.
OSTY_OLD_SAVED_GID	Integer	10	561	570	Old saved GID.
OSTY_TGT_REAL_UID	Integer	10	572	581	Target real UID.
OSTY_TGT_EFF_UID	Integer	10	583	592	Target effective UID.
OSTY_TGT_SAV_UID	Integer	10	594	603	Target saved UID.
OSTY_TGT_PID	Integer	10	605	614	Target process ID.

Event Qualifiers for the Open Slave TTY Record

The event qualifiers that may be associated with the opening of a slave TTY are shown in Table 130 on page 245.

Table 130. Event Code Qualifiers for Open Slave TTY Records

Event Qualifier	Event Number	Event Description
SUCCESS	00	Access allowed.
NOTAUTH	01	Not authorized to the specified process.

The Format of the RACLINK Command Record Extension

Table 131 on page 246 describes the format of a record that is created by the a RACLINK command.

Table 131. Format of the RACLINK Command Record Extension

Field Name	Type	Length	Position		Comments
			Start	End	
RACL_USER_NAME	Char	20	282	301	The name associated with the user ID.
RACL_UTK_ENCR	Yes/No	4	303	306	Is the UTOKEN associated with this user encrypted?
RACL_UTK_PRE19	Yes/No	4	308	311	Is this a token for a release earlier than RACF 1.9?
RACL_UTK_VERPROF	Yes/No	4	313	316	Is the VERIFYX propagation flag set?
RACL_UTK_NJEUNUSR	Yes/No	4	318	321	Is this the NJE undefined user?
RACL_UTK_LOGUSR	Yes/No	4	323	326	Is UAUDIT specified for this user?
RACL_UTK_SPECIAL	Yes/No	4	328	331	Is this a RACF SPECIAL user?
RACL_UTK_DEFAULT	Yes/No	4	333	336	Is this a default token?
RACL_UTK_UNKNUSR	Yes/No	4	338	341	Is this an undefined user?
RACL_UTK_ERROR	Yes/No	4	343	346	Is this user token in error?
RACL_UTK_TRUSTED	Yes/No	4	348	351	Is this user a part of the trusted computing base (TCB)?
RACL_UTK_SESSTYPE	Char	8	353	360	The session type of this session.
RACL_UTK_SURROGAT	Yes/No	4	362	365	Is this a surrogate user?
RACL_UTK_REMOTE	Yes/No	4	367	370	Is this a remote job?
RACL_UTK_PRIV	Yes/No	4	372	375	Is this a privileged user ID?
RACL_UTK_SECL	Char	8	377	384	The SECLABEL of the user.
RACL_UTK_EXECNODE	Char	8	386	393	The execution node of the work.
RACL_UTK_SUSER_ID	Char	8	395	402	The submitting user ID.
RACL_UTK_SNODE	Char	8	404	411	The submitting node.
RACL_UTK_SGRP_ID	Char	8	413	420	The submitting group ID.
RACL_UTK_SPOE	Char	8	422	429	The port of entry.
RACL_UTK_SPCLASS	Char	8	431	438	Class of the POE. Valid values are "TERMINAL", "CONSOLE", "JESINPUT", and "APPCPORT".
RACL_UTK_USER_ID	Char	8	440	447	User ID associated with the record.
RACL_UTK_GRP_ID	Char	8	449	456	Group ID associated with the record.
RACL_UTK_DFT_GRP	Yes/No	4	458	461	Is a default group assigned?
RACL_UTK_DFT_SECL	Yes/No	4	463	466	Is a default SECLABEL assigned?
RACL_PHASE	Char	20	468	487	Phase of this RACF command. Valid values are "LOCAL ISSUANCE", "TARGET PROCESSING", and "TARGET RESPONSE".
RACL_ISSUE_NODE	Char	8	489	496	Node that originated the command.
RACL_ISSUE_ID	Char	8	498	505	User ID that originated the command.
RACL_SOURCE_ID	Char	8	507	514	User ID for the association. From the ID keyword.
RACL_TGT_NODE	Char	8	516	523	Node that is the destination of the command.
RACL_TGT_ID	Char	8	525	532	User ID that is the destination of the command.
RACL_TGT_AUTH_ID	Char	8	534	541	User ID under whose authority the association is established.
RACL_SOURCE_SMFID	Char	4	543	546	SMF system identifier of the system that originated the command.
RACL_SOURCE_TIME	Char	8	548	555	Time that the command originated.

Table 131. Format of the RACLINK Command Record Extension (continued)					
Field Name	**Type**	**Length**	**Position**		**Comments**
			Start	**End**	
RACL_SOURCE_DATE	Char	10	557	566	Date that the command originated.
RACL_PWD_STATUS	Char	8	568	575	Status of the password sent with the command. Valid values are: **SUPPLIED** A password was supplied on a DEFINE command. This value only occurs for a `LOCAL ISSUANCE` phase record. **VALID** The password that was supplied on a DEFINE command is correct. This value only occurs for `TARGET PROCESSING` and `TARGET RESPONSE` phase records. **NOTVALID** The password that was supplied on a DEFINE command is not correct. This value only occurs for a `TARGET PROCESSING` phase record. **NONE** No password was supplied for the DEFINE command. This value can occur for any phase record. A blank value indicates that an UNDEFINE or APPROVE command was issued. Neither of these commands have passwords.
RACL_ASSOC_STATUS	Char	8	577	584	Status of the association. Valid values are "`PENDING`", "`ESTAB`", and "`DELETED`".
RACL_SPECIFIED	Char	1024	586	1609	The keywords specified.

Event Qualifiers for the RACLINK Command Records

The event qualifiers that may be associated with the RACLINK command are shown in Table 132 on page 247.

Table 132. Event Code Qualifiers for RACLINK Command Records		
Event Qualifier	**Event Number**	**Event Description**
SUCCESS	00	Command successful.
INSAUTH	01	Insufficient authority (local issuance only).
--------	02	Reserved.
ALRDYDEF	03	Association already defined.
ALRDYAPP	04	Association already approved.
NOMATCH	05	Association does not match.
NOTEXIST	06	Association does not exist.
INVPSWD	07	Invalid password.

The Format of the IPCCHK Access Record Extension

Table 133 on page 248 describes the format of a record that is created by checking access to an IPC.

Table 133. Format of the IPCCHK Record Extension

Field Name	Type	Length	Position		Comments
			Start	End	
ICHK_CLASS	Char	8	282	289	Class name.
ICHK_USER_NAME	Char	20	291	310	The name associated with the user ID.
ICHK_UTK_ENCR	Yes/No	4	312	315	Is the UTOKEN associated with this user encrypted?
ICHK_UTK_PRE19	Yes/No	4	317	320	Is this a token for a release earlier than RACF 1.9?
ICHK_UTK_VERPROF	Yes/No	4	322	325	Is the VERIFYX propagation flag set?
ICHK_UTK_NJEUNUSR	Yes/No	4	327	330	Is this the NJE undefined user?
ICHK_UTK_LOGUSR	Yes/No	4	332	335	Is UAUDIT specified for this user?
ICHK_UTK_SPECIAL	Yes/No	4	337	340	Is this a RACF SPECIAL user?
ICHK_UTK_DEFAULT	Yes/No	4	342	345	Is this a default token?
ICHK_UTK_UNKNUSR	Yes/No	4	347	350	Is this an undefined user?
ICHK_UTK_ERROR	Yes/No	4	352	355	Is this user token in error?
ICHK_UTK_TRUSTED	Yes/No	4	357	360	Is this user a part of the trusted computing base (TCB)?
ICHK_UTK_SESSTYPE	Char	8	362	369	The session type of this session.
ICHK_UTK_SURROGAT	Yes/No	4	371	374	Is this a surrogate user?
ICHK_UTK_REMOTE	Yes/No	4	376	379	Is this a remote job?
ICHK_UTK_PRIV	Yes/No	4	381	384	Is this a privileged user ID?
ICHK_UTK_SECL	Char	8	386	393	The SECLABEL of the user.
ICHK_UTK_EXECNODE	Char	8	395	402	The execution node of the work.
ICHK_UTK_SUSER_ID	Char	8	404	411	The submitting user ID.
ICHK_UTK_SNODE	Char	8	413	420	The submitting node.
ICHK_UTK_SGRP_ID	Char	8	422	429	The submitting group ID.
ICHK_UTK_SPOE	Char	8	431	438	The port of entry.
ICHK_UTK_SPCLASS	Char	8	440	447	Class of the POE. Valid values are "TERMINAL", "CONSOLE", "JESINPUT", and "APPCPORT".
ICHK_UTK_USER_ID	Char	8	449	456	User ID associated with the record.
ICHK_UTK_GRP_ID	Char	8	458	465	Group ID associated with the record.
ICHK_UTK_DFT_GRP	Yes/No	4	467	470	Is a default group assigned?
ICHK_UTK_DFT_SECL	Yes/No	4	472	475	Is a default SECLABEL assigned?
ICHK_APPC_LINK	Char	16	477	492	Key to link together APPC records.
ICHK_AUDIT_CODE	Char	11	494	504	Audit function code. For more information on the function codes, see Appendix I, "OpenExtensions Audit Function Codes," on page 439.
ICHK_OLD_REAL_UID	Integer	10	506	515	Old real UID.
ICHK_OLD_EFF_UID	Integer	10	517	526	Old effective UID.
ICHK_OLD_SAVED_UID	Integer	10	528	537	Old saved UID.
ICHK_OLD_REAL_GID	Integer	10	539	548	Old real GID.
ICHK_OLD_EFF_GID	Integer	10	550	559	Old effective GID.
ICHK_OLD_SAVED_GID	Integer	10	561	570	Old saved GID.
ICHK_KEY_OWN_UID	Integer	10	572	581	The owner UID associated with the key.

Table 133. Format of the IPCCHK Record Extension (continued)

Field Name	Type	Length	Position		Comments
			Start	End	
ICHK_KEY_OWN_GID	Integer	10	583	592	The owner GID associated with the key.
ICHK_REQUEST_READ	Yes/No	4	594	597	Did the requested access include read?
ICHK_REQUEST_WRITE	Yes/No	4	599	602	Did the requested access include write?
ICHK_REQUEST_EXEC	Yes/No	4	604	607	Did the requested access include execute?
ICHK_RESERVED_01	Yes/No	4	609	612	Reserved.
ICHK_ACCESS_TYPE	Char	8	614	621	What bits were used in granting the access? Valid values are "OWNER", "GROUP", "NO", and "OTHER".
ICHK_ALLOWED_READ	Yes/No	4	623	626	Was read access allowed?
ICHK_ALLOWED_WRITE	Yes/No	4	628	631	Was write access allowed?
ICHK_RESERVED_02	Yes/No	4	633	636	Reserved.
ICHK_KEY	Char	8	638	645	The key of the IPC resource.
ICHK_ID	Integer	10	647	656	The unique decimal identifier of the IPC resource.
ICHK_CREATOR_UID	Integer	10	658	667	The UID of the creator.
ICHK_CREATOR_GID	Integer	10	669	678	The GID of the creator.

Event Qualifiers for IPCCHK Requests

The event qualifiers that may be associated with a check IPC access event are shown in Table 134 on page 249.

Table 134. Event Code Qualifiers for IPCCHK Records

Event Qualifier	Event Number	Event Description
SUCCESS	00	Access allowed.
NOTAUTH	01	Not authorized to the resource.

The Format of the IPCGET Access Record Extension

Table 135 on page 249 describes the format of a record that is created by creating an IPC.

Table 135. Format of the IPCGET Access Record Extension

Field Name	Type	Length	Position		Comments
			Start	End	
IGET_CLASS	Char	8	282	289	Class name.
IGET_USER_NAME	Char	20	291	310	The name associated with the user ID.
IGET_UTK_ENCR	Yes/No	4	312	315	Is the UTOKEN associated with this user encrypted?
IGET_UTK_PRE19	Yes/No	4	317	320	Is this a token for a release earlier than RACF 1.9?
IGET_UTK_VERPROF	Yes/No	4	322	325	Is the VERIFYX propagation flag set?
IGET_UTK_NJEUNUSR	Yes/No	4	327	330	Is this the NJE undefined user?
IGET_UTK_LOGUSR	Yes/No	4	332	335	Is UAUDIT specified for this user?
IGET_UTK_SPECIAL	Yes/No	4	337	340	Is this a RACF SPECIAL user?
IGET_UTK_DEFAULT	Yes/No	4	342	345	Is this a default token?
IGET_UTK_UNKNUSR	Yes/No	4	347	350	Is this an undefined user?

Table 135. Format of the IPCGET Access Record Extension (continued)

Field Name	Type	Length	Position		Comments
			Start	End	
IGET_UTK_ERROR	Yes/No	4	352	355	Is this user token in error?
IGET_UTK_TRUSTED	Yes/No	4	357	360	Is this user a part of the trusted computing base (TCB)?
IGET_UTK_SESSTYPE	Char	8	362	369	The session type of this session.
IGET_UTK_SURROGAT	Yes/No	4	371	374	Is this a surrogate user?
IGET_UTK_REMOTE	Yes/No	4	376	379	Is this a remote job?
IGET_UTK_PRIV	Yes/No	4	381	384	Is this a privileged user ID?
IGET_UTK_SECL	Char	8	386	393	The SECLABEL of the user.
IGET_UTK_EXECNODE	Char	8	395	402	The execution node of the work.
IGET_UTK_SUSER_ID	Char	8	404	411	The submitting user ID.
IGET_UTK_SNODE	Char	8	413	420	The submitting node.
IGET_UTK_SGRP_ID	Char	8	422	429	The submitting group ID.
IGET_UTK_SPOE	Char	8	431	438	The port of entry.
IGET_UTK_SPCLASS	Char	8	440	447	Class of the POE. Valid values are "TERMINAL", "CONSOLE", "JESINPUT", and "APPCPORT".
IGET_UTK_USER_ID	Char	8	449	456	User ID associated with the record.
IGET_UTK_GRP_ID	Char	8	458	465	Group ID associated with the record.
IGET_UTK_DFT_GRP	Yes/No	4	467	470	Is a default group assigned?
IGET_UTK_DFT_SECL	Yes/No	4	472	475	Is a default SECLABEL assigned?
IGET_APPC_LINK	Char	16	477	492	Key to link together APPC records.
IGET_AUDIT_CODE	Char	11	494	504	Audit function code. For more information on the function codes, see Appendix I, “OpenExtensions Audit Function Codes,” on page 439.
IGET_OLD_REAL_UID	Integer	10	506	515	Old real UID.
IGET_OLD_EFF_UID	Integer	10	517	526	Old effective UID.
IGET_OLD_SAVED_UID	Integer	10	528	537	Old saved UID.
IGET_OLD_REAL_GID	Integer	10	539	548	Old real GID.
IGET_OLD_EFF_GID	Integer	10	550	559	Old effective GID.
IGET_OLD_SAVED_GID	Integer	10	561	570	Old saved GID.
IGET_KEY_OWN_UID	Integer	10	572	581	The owner UID associated with the key.
IGET_KEY_OWN_GID	Integer	10	583	592	The owner GID associated with the key.
IGET_RESERVED_01	Yes/No	4	594	597	Reserved.
IGET_RESERVED_02	Yes/No	4	599	602	Reserved.
IGET_RESERVED_03	Yes/No	4	604	607	Reserved.
IGET_REQ_OWN_READ	Yes/No	4	609	612	Was the owner READ bit requested on for this file?
IGET_REQ_OWN_WRITE	Yes/No	4	614	617	Was the owner WRITE bit requested on for this file?
IGET_REQ_OWN_EXEC	Yes/No	4	619	622	Was the owner EXECUTE bit requested on for this file?
IGET_REQ_GRP_READ	Yes/No	4	624	627	Was the group READ bit requested on for this file?
IGET_REQ_GRP_WRITE	Yes/No	4	629	632	Was the group WRITE bit requested on for this file?
IGET_REQ_GRP_EXEC	Yes/No	4	634	637	Was the group EXECUTE bit requested on for this file?

Table 135. Format of the IPCGET Access Record Extension (continued)					
Field Name	**Type**	**Length**	**Position**		**Comments**
			Start	**End**	
IGET_REQ_OTH_READ	Yes/No	4	639	642	Was the other READ bit requested on for this file?
IGET_REQ_OTH_WRITE	Yes/No	4	644	647	Was the other WRITE bit requested on for this file?
IGET_REQ_OTH_EXEC	Yes/No	4	649	652	Was the other EXECUTE bit requested on for this file?
IGET_KEY	Char	8	654	661	The key of the IPC resource.
IGET_ID	Integer	10	663	672	The unique decimal identifier of the IPC resource.
IGET_CREATOR_UID	Integer	10	674	683	The UID of the creator.
IGET_CREATOR_GID	Integer	10	685	694	The GID of the creator.

Event Qualifiers for IPCGET Access Requests

The event qualifiers that may be associated with a make ISP event are shown in Table 136 on page 251.

Table 136. Event Code Qualifiers for IPCGET Acess Records		
Event Qualifier	**Event Number**	**Event Description**
SUCCESS	00	Access allowed.

The Format of the IPCCTL Access Record Extension

Table 137 on page 251 describes the format of a record that is created by the IPCCTL function.

Table 137. Format of the IPCCTL Access Record Extension					
Field Name	**Type**	**Length**	**Position**		**Comments**
			Start	**End**	
ICTL_CLASS	Char	8	282	289	Class name.
ICTL_USER_NAME	Char	20	291	310	The name associated with the user ID.
ICTL_UTK_ENCR	Yes/No	4	312	315	Is the UTOKEN associated with this user encrypted?
ICTL_UTK_PRE19	Yes/No	4	317	320	Is this a token for a release earlier than RACF 1.9?
ICTL_UTK_VERPROF	Yes/No	4	322	325	Is the VERIFYX propagation flag set?
ICTL_UTK_NJEUNUSR	Yes/No	4	327	330	Is this the NJE undefined user?
ICTL_UTK_LOGUSR	Yes/No	4	332	335	Is UAUDIT specified for this user?
ICTL_UTK_SPECIAL	Yes/No	4	337	340	Is this a RACF SPECIAL user?
ICTL_UTK_DEFAULT	Yes/No	4	342	345	Is this a default token?
ICTL_UTK_UNKNUSR	Yes/No	4	347	350	Is this an undefined user?
ICTL_UTK_ERROR	Yes/No	4	352	355	Is this user token in error?
ICTL_UTK_TRUSTED	Yes/No	4	357	360	Is this user a part of the trusted computing base (TCB)?
ICTL_UTK_SESSTYPE	Char	8	362	369	The session type of this session.
ICTL_UTK_SURROGAT	Yes/No	4	371	374	Is this a surrogate user?
ICTL_UTK_REMOTE	Yes/No	4	376	379	Is this a remote job?
ICTL_UTK_PRIV	Yes/No	4	381	384	Is this a privileged user ID?
ICTL_UTK_SECL	Char	8	386	393	The SECLABEL of the user.
ICTL_UTK_EXECNODE	Char	8	395	402	The execution node of the work.

Table 137. Format of the IPCCTL Access Record Extension (continued)

Field Name	Type	Length	Position		Comments
			Start	End	
ICTL_UTK_SUSER_ID	Char	8	404	411	The submitting user ID.
ICTL_UTK_SNODE	Char	8	413	420	The submitting node.
ICTL_UTK_SGRP_ID	Char	8	422	429	The submitting group ID.
ICTL_UTK_SPOE	Char	8	431	438	The port of entry.
ICTL_UTK_SPCLASS	Char	8	440	447	Class of the POE. Valid values are "TERMINAL", "CONSOLE", "JESINPUT", and "APPCPORT".
ICTL_UTK_USER_ID	Char	8	449	456	User ID associated with the record.
ICTL_UTK_GRP_ID	Char	8	458	465	Group ID associated with the record.
ICTL_UTK_DFT_GRP	Yes/No	4	467	470	Is a default group assigned?
ICTL_UTK_DFT_SECL	Yes/No	4	472	475	Is a default SECLABEL assigned?
ICTL_APPC_LINK	Char	16	477	492	Key to link together APPC records.
ICTL_AUDIT_CODE	Char	11	494	504	Audit function code. For more information on the function codes, see Appendix I, “OpenExtensions Audit Function Codes,” on page 439.
ICTL_OLD_REAL_UID	Integer	10	506	515	Old real UID.
ICTL_OLD_EFF_UID	Integer	10	517	526	Old effective UID.
ICTL_OLD_SAVED_UID	Integer	10	528	537	Old saved UID.
ICTL_OLD_REAL_GID	Integer	10	539	548	Old real GID.
ICTL_OLD_EFF_GID	Integer	10	550	559	Old effective GID.
ICTL_OLD_SAVED_GID	Integer	10	561	570	Old saved GID.
ICTL_KEY_OWN_UID	Integer	10	572	581	The owner UID associated with the key
ICTL_KEY_OWN_GID	Integer	10	583	592	The owner GID associated with the key.
ICTL_UID	Integer	10	594	603	The owner UID input parameter.
ICTL_GID	Integer	10	605	614	The owner GID input parameter.
ICTL_RESERVED_01	Yes/No	4	616	619	Reserved.
ICTL_RESERVED_02	Yes/No	4	621	624	Reserved.
ICTL_RESERVED_03	Yes/No	4	626	629	Reserved.
ICTL_OLD_OWN_READ	Yes/No	4	631	634	Was the owner READ bit on for this file?
ICTL_OLD_OWN_WRITE	Yes/No	4	636	639	Was the owner WRITE bit on for this file?
ICTL_OLD_OWN_EXEC	Yes/No	4	641	644	Was the owner EXECUTE bit on for this file?
ICTL_OLD_GRP_READ	Yes/No	4	646	649	Was the group READ bit on for this file?
ICTL_OLD_GRP_WRITE	Yes/No	4	651	654	Was the group WRITE bit on for this file?
ICTL_OLD_GRP_EXEC	Yes/No	4	656	659	Was the group EXECUTE bit on for this file?
ICTL_OLD_OTH_READ	Yes/No	4	661	664	Was the other READ bit on for this file?
ICTL_OLD_OTH_WRITE	Yes/No	4	666	669	Was the other WRITE bit on for this file?
ICTL_OLD_OTH_EXEC	Yes/No	4	671	674	Was the other EXECUTE bit on for this file?
ICTL_RESERVED_04	Yes/No	4	676	679	Reserved.
ICTL_RESERVED_05	Yes/No	4	681	684	Reserved.
ICTL_RESERVED_06	Yes/No	4	686	689	Reserved.
ICTL_NEW_OWN_READ	Yes/No	4	691	694	Is the owner READ bit on for this file?

Table 137. Format of the IPCCTL Access Record Extension (continued)

Field Name	Type	Length	Position		Comments
			Start	End	
ICTL_NEW_OWN_WRITE	Yes/No	4	696	699	Is the owner WRITE bit on for this file?
ICTL_NEW_OWN_EXEC	Yes/No	4	701	704	Is the owner EXECUTE bit on for this file?
ICTL_NEW_GRP_READ	Yes/No	4	706	709	Is the group READ bit on for this file?
ICTL_NEW_GRP_WRITE	Yes/No	4	711	714	Is the group WRITE bit on for this file?
ICTL_NEW_GRP_EXEC	Yes/No	4	716	719	Is the group EXECUTE bit on for this file?
ICTL_NEW_OTH_READ	Yes/No	4	721	724	Is the other READ bit on for this file?
ICTL_NEW_OTH_WRITE	Yes/No	4	726	729	Is the other WRITE bit on for this file?
ICTL_NEW_OTH_EXEC	Yes/No	4	731	734	Is the other EXECUTE bit on for this file?
ICTL_SERVICE_CODE	Char	11	736	746	The service that was being processed.
ICTL_RESERVED_07	Yes/No	4	748	751	Reserved.
ICTL_RESERVED_08	Yes/No	4	753	756	Reserved.
ICTL_RESERVED_09	Yes/No	4	758	761	Reserved.
ICTL_REQ_OWN_READ	Yes/No	4	763	766	Was the owner READ bit requested on for this file?
ICTL_REQ_OWN_WRITE	Yes/No	4	768	771	Was the owner WRITE bit requested on for this file?
ICTL_REQ_OWN_EXEC	Yes/No	4	773	776	Was the owner EXECUTE bit requested on for this file?
ICTL_REQ_GRP_READ	Yes/No	4	778	781	Was the group READ bit requested on for this file?
ICTL_REQ_GRP_WRITE	Yes/No	4	783	786	Was the group WRITE bit requested on for this file?
ICTL_REQ_GRP_EXEC	Yes/No	4	788	791	Was the group EXECUTE bit requested on for this file?
ICTL_REQ_OTH_READ	Yes/No	4	793	796	Was the other READ bit requested on for this file?
ICTL_REQ_OTH_WRITE	Yes/No	4	798	801	Was the other WRITE bit requested on for this file?
ICTL_REQ_OTH_EXEC	Yes/No	4	803	806	Was the other EXECUTE bit requested on for this file?
ICTL_KEY	Char	8	808	815	The key of the IPC resource.
ICTL_ID	Integer	10	817	826	The unique decimal identifier of the IPC resource.
ICTL_CREATOR_UID	Integer	10	828	837	The UID of the creator.
ICTL_CREATOR_GID	Integer	10	839	848	The GID of the creator.

Event Qualifiers for IPCCTL Access Requests

The event qualifiers that may be associated with a IPCCTL event are shown in Table 138 on page 253.

Table 138. Event Code Qualifiers for IPCCTL Access Records

Event Qualifier	Event Number	Event Description
SUCCESS	00	Access allowed.
NOTAUTH	01	Not authorized to the resource.

The Format of the SETGROUP Process Record

Table 139 on page 254 describes the format of a record that is created when a SETGROUP function is performed.

Table 139. Format of the SETGROUP Process Record Extension

Field Name	Type	Length	Position		Comments
			Start	End	
SETG_CLASS	Char	8	282	289	Class name.
SETG_USER_NAME	Char	20	291	310	The name associated with the user ID.
SETG_UTK_ENCR	Yes/No	4	312	315	Is the UTOKEN associated with this user encrypted?
SETG_UTK_PRE19	Yes/No	4	317	320	Is this a token for a release earlier than RACF 1.9?
SETG_UTK_VERPROF	Yes/No	4	322	325	Is the VERIFYX propagation flag set?
SETG_UTK_NJEUNUSR	Yes/No	4	327	330	Is this the NJE undefined user?
SETG_UTK_LOGUSR	Yes/No	4	332	335	Is UAUDIT specified for this user?
SETG_UTK_SPECIAL	Yes/No	4	337	340	Is this a RACF SPECIAL user?
SETG_UTK_DEFAULT	Yes/No	4	342	345	Is this a default token?
SETG_UTK_UNKNUSR	Yes/No	4	347	350	Is this an undefined user?
SETG_UTK_ERROR	Yes/No	4	352	355	Is this user token in error?
SETG_UTK_TRUSTED	Yes/No	4	357	360	Is this user a part of the trusted computing base (TCB)?
SETG_UTK_SESSTYPE	Char	8	362	369	The session type of this session.
SETG_UTK_SURROGAT	Yes/No	4	371	374	Is this a surrogate user?
SETG_UTK_REMOTE	Yes/No	4	376	379	Is this a remote job?
SETG_UTK_PRIV	Yes/No	4	381	384	Is this a privileged user ID?
SETG_UTK_SECL	Char	8	386	393	The SECLABEL of the user.
SETG_UTK_EXECNODE	Char	8	395	402	The execution node of the work.
SETG_UTK_SUSER_ID	Char	8	404	411	The submitting user ID.
SETG_UTK_SNODE	Char	8	413	420	The submitting node.
SETG_UTK_SGRP_ID	Char	8	422	429	The submitting group ID.
SETG_UTK_SPOE	Char	8	431	438	The port of entry.
SETG_UTK_SPCLASS	Char	8	440	447	Class of the POE. Valid values are "TERMINAL", "CONSOLE", "JESINPUT", and "APPCPORT".
SETG_UTK_USER_ID	Char	8	449	456	User ID associated with the record.
SETG_UTK_GRP_ID	Char	8	458	465	Group ID associated with the record.
SETG_UTK_DFT_GRP	Yes/No	4	467	470	Is a default group assigned?
SETG_UTK_DFT_SECL	Yes/No	4	472	475	Is a default SECLABEL assigned?
SETG_APPC_LINK	Char	16	477	492	A key to link together audit records for a user's APPC transaction processing work.
SETG_AUDIT_CODE	Char	11	494	504	Audit function code. For more information on the function codes, see Appendix I, “OpenExtensions Audit Function Codes,” on page 439.
SETG_OLD_REAL_UID	Integer	10	506	515	Old real UID.
SETG_OLD_EFF_UID	Integer	10	517	526	Old effective UID.
SETG_OLD_SAVED_UID	Integer	10	528	537	Old saved UID.
SETG_OLD_REAL_GID	Integer	10	539	548	Old real GID.
SETG_OLD_EFF_GID	Integer	10	550	559	Old effective GID.
SETG_OLD_SAVED_GID	Integer	10	561	570	Old saved GID.

Table 139. Format of the SETGROUP Process Record Extension (continued)

Field Name	Type	Length	Position		Comments
			Start	End	
SETG_DCE_LINK	Char	16	572	587	Link to connect DCE records that originate from a single DCE request.
SETG_AUTH_TYPE	Char	13	589	601	Defines the type of request. Valid values are: "SERVER" and "AUTH_CLIENT" and "UNAUTH_CLIENT".

Event Qualifiers for the SETGROUP Process Record Extension

The event qualifiers that may be associated with the SETGROUP function are shown in Table 140 on page 255.

Table 140. Event Code Qualifiers for SETGROUP Process Records

Event Qualifier	Event Number	Event Description
SUCCESS	00	Process successfully initialized.
NOTAUTH	01	User does not have the super user authority.

The Format of the Check Owner, Two Files Record Extension

Table 141 on page 255 describes the format of a record that is created by the check owner of two files function.

Table 141. Format of the Check Owner, Two Files Record Extension

Field Name	Type	Length	Position		Comments
			Start	End	
CKO2_CLASS	Char	8	282	289	Class name.
CKO2_USER_NAME	Char	20	291	310	The name associated with the user ID.
CKO2_UTK_ENCR	Yes/No	4	312	315	Is the UTOKEN associated with this user encrypted?
CKO2_UTK_PRE19	Yes/No	4	317	320	Is this a token for a release earlier than RACF 1.9?
CKO2_UTK_VERPROF	Yes/No	4	322	325	Is the VERIFYX propagation flag set?
CKO2_UTK_NJEUNUSR	Yes/No	4	327	330	Is this the NJE undefined user?
CKO2_UTK_LOGUSR	Yes/No	4	332	335	Is UAUDIT specified for this user?
CKO2_UTK_SPECIAL	Yes/No	4	337	340	Is this a RACF SPECIAL user?
CKO2_UTK_DEFAULT	Yes/No	4	342	345	Is this a default token?
CKO2_UTK_UNKNUSR	Yes/No	4	347	350	Is this an undefined user?
CKO2_UTK_ERROR	Yes/No	4	352	355	Is this user token in error?
CKO2_UTK_TRUSTED	Yes/No	4	357	360	Is this user a part of the trusted computing base (TCB)?
CKO2_UTK_SESSTYPE	Char	8	362	369	The session type of this session.
CKO2_UTK_SURROGAT	Yes/No	4	371	374	Is this a surrogate user?
CKO2_UTK_REMOTE	Yes/No	4	376	379	Is this a remote job?
CKO2_UTK_PRIV	Yes/No	4	381	384	Is this a privileged user ID?
CKO2_UTK_SECL	Char	8	386	393	The SECLABEL of the user.
CKO2_UTK_EXECNODE	Char	8	395	402	The execution node of the work.
CKO2_UTK_SUSER_ID	Char	8	404	411	The submitting user ID.

Table 141. Format of the Check Owner, Two Files Record Extension (continued)

Field Name	Type	Length	Position		Comments
			Start	End	
CKO2_UTK_SNODE	Char	8	413	420	The submitting node.
CKO2_UTK_SGRP_ID	Char	8	422	429	The submitting group ID.
CKO2_UTK_SPOE	Char	8	431	438	The port of entry.
CKO2_UTK_SPCLASS	Char	8	440	447	Class of the POE. Valid values are "TERMINAL", "CONSOLE", "JESINPUT", and "APPCPORT".
CKO2_UTK_USER_ID	Char	8	449	456	User ID associated with the record.
CKO2_UTK_GRP_ID	Char	8	458	465	Group ID associated with the record.
CKO2_UTK_DFT_GRP	Yes/No	4	467	470	Is a default group assigned?
CKO2_UTK_DFT_SECL	Yes/No	4	472	475	Is a default SECLABEL assigned?
CKO2_APPC_LINK	Char	16	477	492	Key to link together APPC records.
CKO2_AUDIT_CODE	Char	11	494	504	Audit function code. For more information on the function codes, see Appendix I, “OpenExtensions Audit Function Codes,” on page 439.
CKO2_OLD_REAL_UID	Integer	10	506	515	Old real UID.
CKO2_OLD_EFF_UID	Integer	10	517	526	Old effective UID.
CKO2_OLD_SAVED_UID	Integer	10	528	537	Old saved UID.
CKO2_OLD_REAL_GID	Integer	10	539	548	Old real GID.
CKO2_OLD_EFF_GID	Integer	10	550	559	Old effective GID.
CKO2_OLD_SAVED_GID	Integer	10	561	570	Old saved GID.
CKO2_PATH_NAME	Char	1023	572	1594	The requested path name.
CKO2_FILE1_ID	Char	32	1596	1627	First file ID.
CKO2_FILE1_OWN_UID	Integer	10	1629	1638	The owner UID associated with the first file.
CKO2_FILE1_OWN_GID	Integer	10	1640	1649	The owner GID associated with the first file.
CKO2_FILE2_ID	Char	32	1651	1682	Second requested file ID.
CKO2_FILE2_OWN_UID	Integer	10	1684	1693	UID of the owner of the second file.
CKO2_FILE2_OWN_GID	Integer	10	1695	1704	GID of the owner of the second file.
CKO2_DCE_LINK	Char	16	1706	1721	Link to connect DCE records that originate from a single DCE request.
CKO2_AUTH_TYPE	Char	13	1723	1735	Defines the type of request. Valid values are: "SERVER" and "AUTH_CLIENT" and "UNAUTH_CLIENT".

Event Qualifiers for Check Owner, Two Files Access Requests

The event qualifiers that may be associated with a check file owner (two files) event are shown in Table 142 on page 256.

Table 142. Event Code Qualifiers for Check Owner, Two Files Record

Event Qualifier	Event Number	Event Description
OWNER	00	The user is the owner.
NOTOWNER	01	The user is not the owner.

The Format of the Unloaded SMF Type 81 Data

RACF writes a type 81 record at the completion of the initialization of RACF. Table 143 on page 257 describes the format of the unloaded version of this record.

Table 143. Format of the Unloaded SMF Type 81 Records

Field Name	Type	Length	Position		Comments
			Start	End	
RINI_EVENT_TYPE	Char	8	1	8	The type of the event. Set to "RACFINIT".
RINI_RESERVED_01	Char	8	10	17	This field is reserved and is set to blanks to allow a common alignment with other unloaded SMF records.
RINI_TIME_WRITTEN	Time	8	19	26	Time that the record was written to SMF.
RINI_DATE_WRITTEN	Date	10	28	37	Date that the record was written to SMF.
RINI_SYSTEM_SMFID	Char	4	39	42	SMF system ID of the system from which the record originates.
RINI_DATASET_NAME	Char	44	44	87	Name of the RACF data base for this IPL.
RINI_DATASET_VOL	Char	6	89	94	Volume upon which the RACF dataset resides
RINI_DATASET_UNIT	Char	3	96	98	Unit name of the RACF database.
RINI_UADS_NAME	Char	44	100	143	Name of the user attribute data set for this IPL.
RINI_UADS_VOL	Char	6	145	150	Volume upon which the user attribute data set resides.
RINI_RACINIT_STATS	Yes/No	4	152	155	Are RACROUTE REQUEST=VERIFY statistics recorded?
RINI_DATASET_STATS	Yes/No	4	157	160	Are dataset statistics recorded?
RINI_RACINIT_PRE	Yes/No	4	162	165	Is there a RACROUTE REQUEST=VERIFY preprocessing exit (ICHRIX01)?
RINI_RACHECK_PRE	Yes/No	4	167	170	Is there a RACROUTE REQUEST=AUTH preprocessing exit (ICHRCX01)?
RINI_RACDEF_PRE	Yes/No	4	172	175	Is there a RACROUTE REQUEST=DEFINE preprocessing exit (ICHRDX01)?
RINI_RACINIT_POST	Yes/No	4	177	180	Is there a RACROUTE REQUEST=VERIFY postprocessing exit (ICHRIN02)?
RINI_RACHECK_POST	Yes/No	4	182	185	Is there a RACROUTE REQUEST=AUTH postprocessing exit (ICHRCX02)?
RINI_NEW_PWD_EXIT	Yes/No	4	187	190	Is there a new password exit routine (ICHPWX01)?
RINI_TAPEVOL_STATS	Yes/No	4	192	195	Are tape volume statistics recorded?
RINI_DASD_STATS	Yes/No	4	197	200	Are DASD statistics recorded?
RINI_TERM_STATS	Yes/No	4	202	205	Are terminal statistics recorded?
RINI_CMD_EXIT	Yes/No	4	207	210	Is the command exit routine ICHCNX00 active? ICHCNX00 is invoked for RACF commands, the IRRUT100 utility, and by IRRXT00 when a RACROUTE REQUEST=EXTRACT is issued for CLASS=DATASET.
RINI_DEL_CMD_EXIT	Yes/No	4	212	215	Is the command exit routine ICHCCX00 active? ICHCCX00 is invoked for the DELGROUP, DELUSER, and REMOVE commands.
RINI_ADSP	Yes/No	4	217	220	Is ADSP active?
RINI_ENCRYPT_EXIT	Yes/No	4	222	225	Is the encryption exit ICHDEX01 active?
RINI_NAMING_CONV	Yes/No	4	227	230	Is the naming convention table ICHNCV00 present?
RINI_TAPEVOL	Yes/No	4	232	235	Is tape volume protection in effect?

Table 143. Format of the Unloaded SMF Type 81 Records (continued)

Field Name	Type	Length	Position		Comments
			Start	End	
RINI_DUP_DSNS	Yes/No	4	237	240	Are duplicate data set names allowed to be defined?
RINI_DASD	Yes/No	4	242	245	Is DASD volume protection in effect?
RINI_FRACHECK_PRE	Yes/No	4	247	250	Is the RACROUTE REQUEST=FASTAUTH preprocessing exit (ICHRFX01) active?
RINI_RACLIST_PRE	Yes/No	4	252	255	Is the RACROUTE REQUEST=LIST pre/postprocessing exit (ICHRLX01) active?
RINI_RACLIST_SEL	Yes/No	4	257	260	Is the RACROUTE REQUEST=LIST selection exit (ICHRLX02) active?
RINI_RACDEF_POST	Yes/No	4	262	265	Is the RACROUTE REQUEST=DEFINE postprocessing exit (ICHRDX02) active?
RINI_AUDIT_USER	Yes/No	4	267	270	Are user class profile changes being audited?
RINI_AUDIT_GROUP	Yes/No	4	272	275	Are group class profile changes being audited?
RINI_AUDIT_DATASET	Yes/No	4	277	280	Are dataset class profile changes being audited?
RINI_AUDIT_TAPEVOL	Yes/No	4	282	285	Are tape volume class profile changes being audited?
RINI_AUDIT_DASDVOL	Yes/No	4	287	290	Are DASD volume class profile changes being audited?
RINI_AUDIT_TERM	Yes/No	4	292	295	Are terminal class profile changes being audited?
RINI_AUDIT_CMDVIOL	Yes/No	4	297	300	Are command violations being audited?
RINI_AUDIT_SPECIAL	Yes/No	4	302	305	Are special users being audited?
RINI_AUDIT_OPER	Yes/No	4	307	310	Are operations users being audited?
RINI_AUDIT_LEVEL	Yes/No	4	312	315	Is auditing by security level in effect?
RINI_ACEE_COMPRESS	Yes/No	4	317	320	Is the IRRACX01 exit in effect?
RINI_FASTAUTH_PRE	Yes/No	4	322	325	Is the FASTAUTH AR mode preprocessing exit (ICHRFX03) in effect?
RINI_FASTAUTH_POST	Yes/No	4	327	330	Is the FASTAUTH AR mode postprocessing exit (ICHRFX04) in effect?
RINI_TERM	Yes/No	4	332	335	Is terminal authorization checking in effect?
RINI_TERM_NONE	Yes/No	4	337	340	Are undefined terminals treated as UACC=NONE?
RINI_REALDSN	Yes/No	4	342	345	Is REALDSN in effect?
RINI_XBMALLRACF	Yes/No	4	347	350	Is the JES XBMALLRACF option in effect?
RINI_EARLYVERIFY	Yes/No	4	352	355	Is the JES EARLYVERIFY option in effect?
RINI_BATCHALLRACF	Yes/No	4	357	360	Is the JES BATCHALLRACF option in effect?
RINI_FRACHECK_POST	Yes/No	4	362	365	Is the RACROUTE REQUEST=FASTAUTH post processing exit (ICHRFX02) in effect?
RINI_PWD_INT	Integer	3	367	369	The maximum password interval.
RINI_SINGLE_DSN	Char	8	371	378	The single level data set name.
RINI_TAPEDSN	Yes/No	4	380	383	Is TAPEDSN in effect?
RINI_PROTECTALL	Yes/No	4	385	388	Is PROTECTALL in effect?
RINI_PROTECTALL_W	Yes/No	4	390	393	Is PROTECTALL warning in effect?
RINI_ERASE	Yes/No	4	395	398	Is ERASE-ON-SCRATCH in effect?
RINI_ERASE_LEVEL	Yes/No	4	400	403	Is ERASE-ON-SCRATCH based on security level in effect?
RINI_ERASE_ALL	Yes/No	4	405	408	Is ERASE-ON-SCRATCH for all data sets in effect?

Table 143. Format of the Unloaded SMF Type 81 Records (continued)

Field Name	Type	Length	Position		Comments
			Start	End	
RINI_EGN	Yes/No	4	410	413	Is enhanced generic naming in effect?
RINI_WHEN_PROGRAM	Yes/No	4	415	418	Is access control by program in effect?
RINI_RETENTION	Integer	5	420	424	System retention period.
RINI_LEVEL_ERASE	Integer	5	426	430	Security level for ERASE-ON-SCRATCH.
RINI_LEVEL_AUDIT	Integer	5	432	436	Security level for auditing.
RINI_SECL_CTRL	Yes/No	4	438	441	Is SECLABELCONTROL in effect?
RINI_CATDSNS	Yes/No	4	443	446	Is CATDSNS in effect?
RINI_MLQUIET	Yes/No	4	448	451	Is MLQUIET in effect?
RINI_MLSTABLE	Yes/No	4	453	456	Is MLSTABLE in effect?
RINI_MLS	Yes/No	4	458	461	Is MLS in effect?
RINI_MLACTIVE	Yes/No	4	463	466	Is MLACTIVE in effect?
RINI_GENERIC_OWNER	Yes/No	4	468	471	Is GENERICOWNER in effect?
RINI_SECL_AUDIT	Yes/No	4	473	476	Is SECLABELAUDIT in effect?
RINI_SESSION_INT	Integer	5	478	482	Partner LU-verification session key interval.
RINI_NJE_NAME_ID	Char	8	484	491	JES NJE name user ID.
RINI_NJE_UDFND_ID	Char	8	493	500	JES UNDEFINEDUSERID user ID.
RINI_COMPATMODE	Yes/No	4	502	505	Is COMPATMODE in effect?
RINI_CATDSNS_FAIL	Yes/No	4	507	510	Is CATDSNS failures in effect?
RINI_MLS_FAIL	Yes/No	4	512	515	Is MLS failures in effect?
RINI_MLACTIVE_FAIL	Yes/No	4	517	520	Is MLACTIVE failures in effect?
RINI_APPLAUD	Yes/No	4	522	525	Is APPLAUDIT in effect?
RINI_DFT_PRI	Char	3	527	529	Default primary language for the installation.
RINI_DFT_SEC	Char	3	531	533	Default secondary language for the installation.
RINI_RESERVED_02	Char	4	535	538	Reserved for IBM's use
RINI_ALL_CMD_EXIT	Yes/No	4	540	543	Did the exit for all commands (IRREVX01) have any active exit routines at IPL time?
RINI_NOADDCREATOR	Yes/No	4	545	548	Is the SETROPTS NOADDCREATOR option in effect?
RINI_ACEE_COMP_XM	Yes/No	4	550	553	Is the IRRACX02 exit in effect?
RINI_ENCRYPT_EXIT2	Yes/No	4	555	558	Is IRRDEX11 exit in effect?
RINI_PWD_HIST	integer	3	560	562	The password history value.
RINI_PWD_REVOKE	integer	3	564	566	The number of incorrect logon passwords before users are revoked.
RINI_PWD_WARN	integer	3	568	570	The number of days before password expiry during which users receive a warning message.
RINI_PWDRULE1_MIN	integer	1	572	572	Password syntax rule 1 minimum length.
RINI_PWDRULE1_MAX	integer	1	574	574	Password syntax rule 1 maximum length.
RINI_PWDRULE1	Char	8	576	583	Password syntax rule 1.
RINI_PWDRULE2_MIN	integer	1	585	585	Password syntax rule 2 minimum length.
RINI_PWDRULE2_MAX	integer	1	587	587	Password syntax rule 2 maximum length.
RINI_PWDRULE2	Char	8	589	596	Password syntax rule 2.

Table 143. Format of the Unloaded SMF Type 81 Records (continued)

Field Name	Type	Length	Position		Comments
			Start	End	
RINI_PWDRULE3_MIN	integer	1	598	598	Password syntax rule 3 minimum length.
RINI_PWDRULE3_MAX	integer	1	600	600	Password syntax rule 3 maximum length.
RINI_PWDRULE3	Char	8	602	609	Password syntax rule 3.
RINI_PWDRULE4_MIN	integer	1	611	611	Password syntax rule 4 minimum length.
RINI_PWDRULE4_MAX	integer	1	613	613	Password syntax rule 4 maximum length.
RINI_PWDRULE4	Char	8	615	622	Password syntax rule 4.
RINI_PWDRULE5_MIN	integer	1	624	624	Password syntax rule 5 minimum length.
RINI_PWDRULE5_MAX	integer	1	626	626	Password syntax rule 5 maximum length.
RINI_PWDRULE5	Char	8	628	635	Password syntax rule 5.
RINI_PWDRULE6_MIN	integer	1	637	637	Password syntax rule 6 minimum length.
RINI_PWDRULE6_MAX	integer	1	639	639	Password syntax rule 6 maximum length.
RINI_PWDRULE6	Char	8	641	648	Password syntax rule 6.
RINI_PWDRULE7_MIN	integer	1	650	650	Password syntax rule 7 minimum length.
RINI_PWDRULE7_MAX	integer	1	652	652	Password syntax rule 7 maximum length.
RINI_PWDRULE7	Char	8	654	661	Password syntax rule 7.
RINI_PWDRULE8_MIN	integer	1	663	663	Password syntax rule 8 minimum length.
RINI_PWDRULE8_MAX	integer	1	665	665	Password syntax rule 8 maximum length.
RINI_PWDRULE8	Char	8	667	674	Password syntax rule 8.
RINI_INACTIVE	integer	3	676	678	The number of days of inactivity before users are revoked.
RINI_GRPLIST	Yes/No	4	680	683	Is list-of-groups processing in effect?
RINI_MODEL_GDG	Yes/No	4	685	688	Is MODEL(GDG) in effect?
RINI_MODEL_USER	Yes/No	4	690	693	Is MODEL(USER) in effect?
RINI_MODEL_GROUP	Yes/No	4	695	698	Is MODEL(GROUP) in effect?
RINI_RSWI_INST_PWD	Yes/No	4	700	703	"Yes" if an installation-defined RVARY SWITCH password is in effect. "No" if the default RVARY SWITCH password is in effect.
RINI_RSTA_INST_PWD	Yes/No	4	705	708	"Yes" if an installation-defined RVARY STATUS password is in effect. "No" if the default RVARY STATUS password is in effect.
RINI_KERBLVL	integer	3	710	712	Level of KERB segment processing in effect.
RINI_MLFS	Char	8	714	721	What is the status of the MLFSOBJ SETROPTS option? (Active or inactive.)
RINI_MLIPC	Char	8	723	730	What is the status of the SETROPTS MLIPCOBJ option? (Active or inactive)
RINI_MLNAMES	Yes/No	4	732	735	Is MLNAMES in effect?
RINI_SLBYSYS	Yes/No	4	737	740	Is SECLBYSYSTEM in effect?
RINI_PWD_MIN	Integer	3	742	744	The password minimum change interval
RINI_PWD_MIXED	Yes/No	4	746	749	Are mixed case passwords allowed?
RINI_NEW_PHR_EXIT	Yes/No	4	751	754	Is the ICHPWX11 exit in effect?
RINI_FLD_VAL_EXIT	Yes/No	4	756	759	Did the field validation exit for custom fields (IRRVAF01) have any active exit routines at IPL time?

Table 143. Format of the Unloaded SMF Type 81 Records (continued)

Field Name	Type	Length	Position		Comments
			Start	End	
RINI_PWD_SPECIAL	Yes/No	4	761	764	Are special characters allowed in passwords?
RINI_PWD_ALG	Char	10	766	777	Algorithm that is used to encrypt passwords and password phrases. Possible values are "KDFAES" and "LEGACY".
RINI_VMX_CON	Char	8	783	790	VMXEVENT control profile in effect
RINI_VMX_AUD	Char	8	792	799	VMXEVENT audit profile in effect

The Format of the Unloaded SMF Type 81 Class Data

Table 144 on page 261 describes the format of the class information that is contained in the SMF type 81 record.

Table 144. Format of the Unloaded SMF Type 81 Class Records

Field Name	Type	Length	Position		Comments
			Start	End	
RINC_EVENT_TYPE	Char	8	1	8	The type of the event. Set to "CLASNAME".
RINC_RESERVED_01	Char	8	10	17	This field is reserved and is set to blanks to allow a common alignment with other unloaded SMF records.
RINC_TIME_WRITTEN	Time	8	19	26	Time that the record was written to SMF.
RINC_DATE_WRITTEN	Date	10	28	37	Date that the record was written to SMF.
RINC_SYSTEM_SMFID	Char	4	39	42	SMF system ID of the system from which the record originates.
RINC_CLASS_NAME	Char	8	44	51	The name of the class.
RINC_STATS	Yes/No	4	53	56	Are statistics collected for this class?
RINC_AUDIT	Yes/No	4	58	61	Is this class being audited?
RINC_ACTIVE	Yes/No	4	63	66	Is this class active?
RINC_GENERIC	Yes/No	4	68	71	May generic profiles be defined in this class?
RINC_GENCMD	Yes/No	4	73	76	Is generic command processing enabled for this class?
RINC_GLOBAL	Yes/No	4	78	81	Is this class enabled for global access checking?
RINC_RACLIST	Yes/No	4	83	86	May this class be RACLISTed?
RINC_GENLIST	Yes/No	4	88	91	May this class be GENLISTed?
RINC_LOG_OPTIONS	Char	8	93	100	The LOGOPTIONS for the class. Valid values are "ALWAYS", "NEVER", and "SUCCESS", and "FAILURES", and "DEFAULT".

The Format of the Unloaded SMF Type 83 Data

The format of unloaded SMF Type 83 records, subtype 3 and above, is described in product-specific documentation.

XML grammar

The RACF SMF data unload utility can generate an eXtensible Markup Language (XML) document containing the SMF data. The names of the tags and the syntax of the values are defined in an XML

schema document, which is used to validate the data contained in an instance document. The RACF schema document is shipped on the RACF service machine's 305 disk as IRRSCHEM SAMPLE.

In general, the RACF XML tag names are derived from the field names used in the tabular output.

Steps for converting RACF field names to XML tag names

Before you begin: You need to know the name of the RACF field name that you want to convert.

Perform the following steps to convert a RACF field name from the tabular format produced by the RACF SMF data unload utility to an XML tag name.

1. Remove the column name and the first "_" character from the field name.
2. Capitalize the first character after each remaining "_" character in the field name. Change all other characters to lowercase.
3. Remove all remaining "_" characters.

When you are done, you have the name of the XML tag that corresponds with the field name that you started with.

Exceptions to this procedure are:

Field name	XML tag name
RINI_TERM	riniTerm
SECL_LINK	link
CAUD_REQUEST_WRITE	caudRequestWrite
CAUD_REQUEST_READ	caudRequestRead
CAUD_REQUEST_EXEC	caudRequestExec
SSCL_OLDSECL	oldSecl
<*col*>_logstring	logstr
KTKT_PRINCIPAL	kerbPrincipal
PDAC_PRINCIPAL	pdasPrincipal
any field with RESERVED in the name	These fields have no XML tag
ACC_NAME	profileName
APPC_NAME	profileName

Example: Converting the field name INIT_USER_NAME to an XML tag name:

1. Start with INIT_USER_NAME

   ```
   INIT_USER_NAME
   ```

2. Remove the column name (INIT) and the first "_" character from the field name.

   ```
   USER_NAME
   ```

3. Capitalize the first character after each remaining "_" character in the field name. Change all other characters to lowercase.

   ```
   user_Name
   ```

4. Remove all remaining "_" characters.

   ```
   userName
   ```

Chapter 5. RACF Database Unload Utility (IRRDBU00)

Note to Reader
If you are sharing a database, RACF recommends running utilities, including the RACF database unload utility, from the uplevel system. If you run this utility and receive record types that are not documented here, check the current documentation for RACF on z/OS.

IRRDBU00 Record Types

The database unload utility gives every record it creates a record type. This record type is a 4-byte identification number located in the first four positions of every record.

The record types and their associated names are:

Record Type
Record Name

0100
Group Basic Data

0101
Group Subgroups

0102
Group Members

0103
Group Installation Data

0110
Group DFP Data

0130
Group OVM Data

0200
User Basic Data

0201
User Categories

0202
User Classes

0203
User Group Connections

0204
User Installation Data

0205
User Connect Data

0210
User DFP Data

0220
User TSO Data

0230
User CICS Data

0231
User CICS Operator Classes

0240
User Language Data

0250
User OPERPARM Data

0251
User OPERPARM Scope

0260
User WORKATTR Data

02A0
User OVM Data

0400
Data Set Basic Data

0401
Data Set Categories

0402
Data Set Conditional Access

0403
Data Set Volumes

0404
Data Set Access

0405
Data Set Installation Data

0410
Data Set DFP Data

0500
General Resource Basic Data

0501
General Resource Tape Volume Data

0502
General Resource Categories

0503
General Resource Members

0504
General Resource Volumes

0505
General Resource Access

0506
General Resource Installation Data

0507
General Resource Conditional Access

0510
General Resource Session Data

0511
General Resource Session Entities

0520
General Resource DLF Data

0521
General Resource DLF Job Names

The record type identification number is in the format **PPSF**, where

PP
Profile type

01
For groups

02
For users

04
For data sets

05
For general resources

S
Segment number

0
Base segment

all others
Segment value determined by the position of the segment in the template

F
Repeat group within the segment. A zero (0) indicates the non-repeat groups within a segment.

The Relationships among Unloaded Database Records

The following figures describe how the records produced by the database unload utility relate to each other. The conventions used in the figures are:

- Only fields showing a relationship to another record type are described
- A line shows a relationship between different types of records
- The complete field names are in the format

  ```
  prefix_fieldname
  ```

  where *prefix* is the unique record prefix assigned to the record and *fieldname* identifies the field in the record. Each section provides the prefix added to the field names.
- The arrows on the connecting line clarify the relationship

  The arrows point to the field that had to have existed first in the RACF database.

  For example, there is a user named GARREN. GARREN creates a group named TEST. The user ID named GARREN had to exist before the group TEST was created.

  In terms of the output from database unload, there exists a user basic data record with GARREN in the USBD_NAME field. There also exists a group basic data record with TEST in the GPBD_NAME field and GARREN in the GPBD_OWNER_ID field.

For more information, see:

- Group records, see “Group Records” on page 265
- User records, see “User Records” on page 267
- Data Set records, see “Data Set Records” on page 269
- General Resource records, see “General Resource Records” on page 270.

Group Records

The prefix representing the record identifier is omitted in the pictorial diagrams. For group records, the prefixes are:

Record Name	Record Type	Record Prefix
Group Basic Data	0100	GPBD
Group Subgroups	0101	GPSGRP
Group Members	0102	GPMEM
Group Installation Data	0103	GPINSTD
Group DFP Data	0110	GPDFP
Group OVM Data	0130	GPOVM

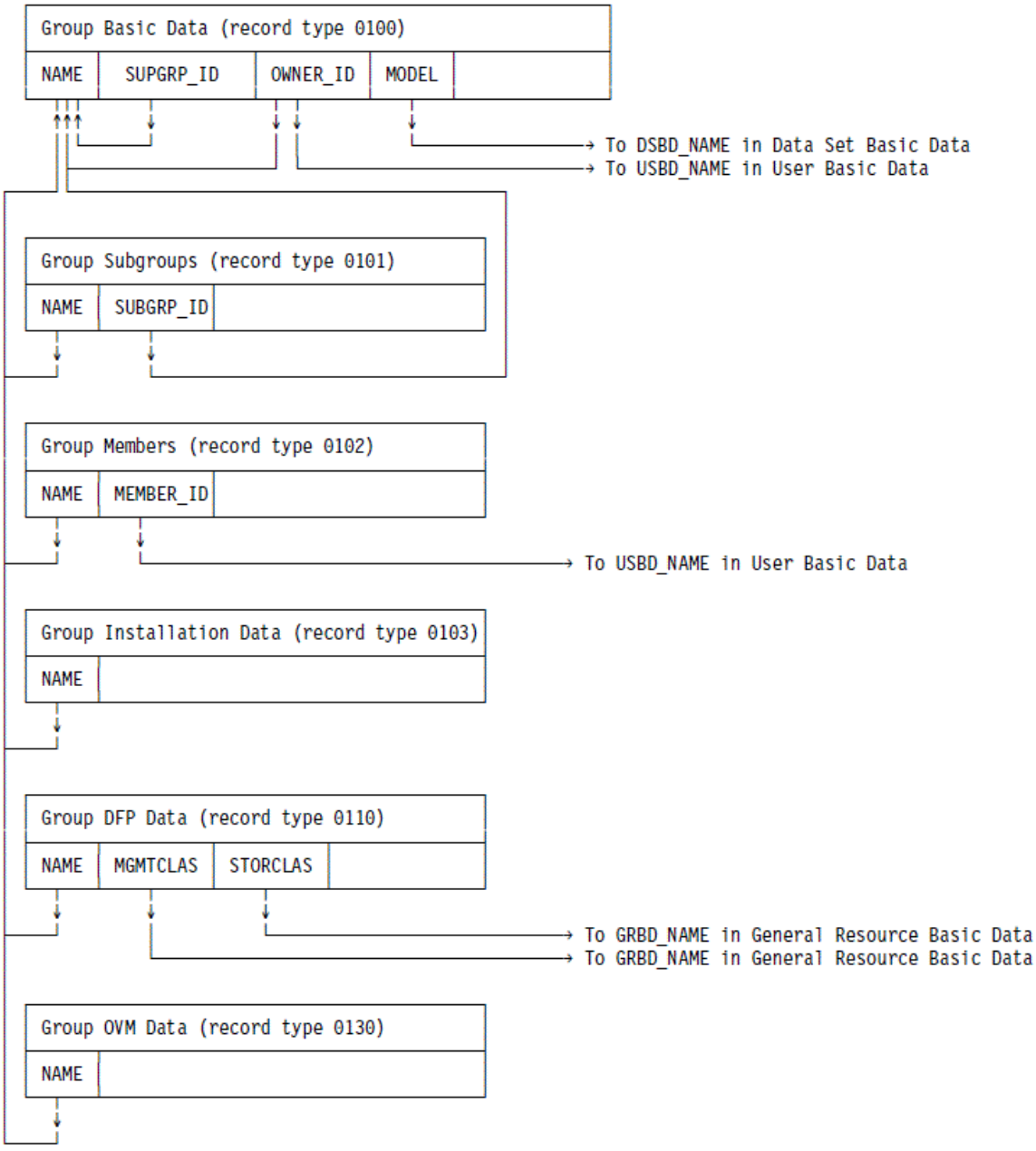


Figure 2. Relationship among the Group Record Types

The records and their relationships are as follows:

Group Basic Data (record type 0100)
Has the following fields and pointers:

- NAME
- SUPGRP_ID, which points to NAME
- OWNER_ID, which points to NAME and to USBD_NAME in User Basic Data
- MODEL, which points to DSBD_NAME in Data Set Basic Data.

Group Subgroups (record type 0101)
Has the following fields and pointers:

- NAME, which points to NAME in Group Basic Data (record type 0100)
- SUBGRP_ID, which points to NAME in Group Basic Data (record type 0100).

Group Members (record type 0102)
Has the following fields and pointers:

- NAME, which points to NAME in Group Basic Data (record type 0100)
- MEMBER_ID, which points to USBD_NAME in User Basic Data.

Group Installation Data (record type 0103)
Has the following field and pointer:

- NAME, which points to NAME in Group Basic Data (record type 0100).

Group DFP Data (record type 0110)
Has the following fields and pointers:

- NAME, which points to NAME in Group Basic Data (record type 0100)
- MGMTCLAS, which points to GRBD_NAME in General Resource Basic Data
- STORCLAS, which points to GRBD_NAME in General Resource Basic Data.

Group OVM Data (record type 0130)
Has the following field and pointer:

- NAME, which points to NAME in Group Basic Data (record type 0100).

User Records

The high level qualifier which represents the table identifier is omitted. For user records, these qualifiers are:

Record Name	Record Type	Record Prefix
User Basic Data	0200	USBD
User Categories	0201	USCAT
User Classes	0202	USCLA
User Group Connections	0203	USGCON
User Installation Data	0204	USINSTD
User Connect Data	0205	USCON
User DFP Data	0210	USDFP
User TSO Data	0220	USTSO
User CICS Data	0230	USCICS
User CICS Operator Classes	0231	USCOPC
User Language Data	0240	USLAN
User OPERPARM Data	0250	USOPR
User OPERPARM Scope	0251	USOPRP

Record Name	Record Type	Record Prefix
User WORKATTR Data	0260	USWRK
User OVM Data	02A0	USOVM

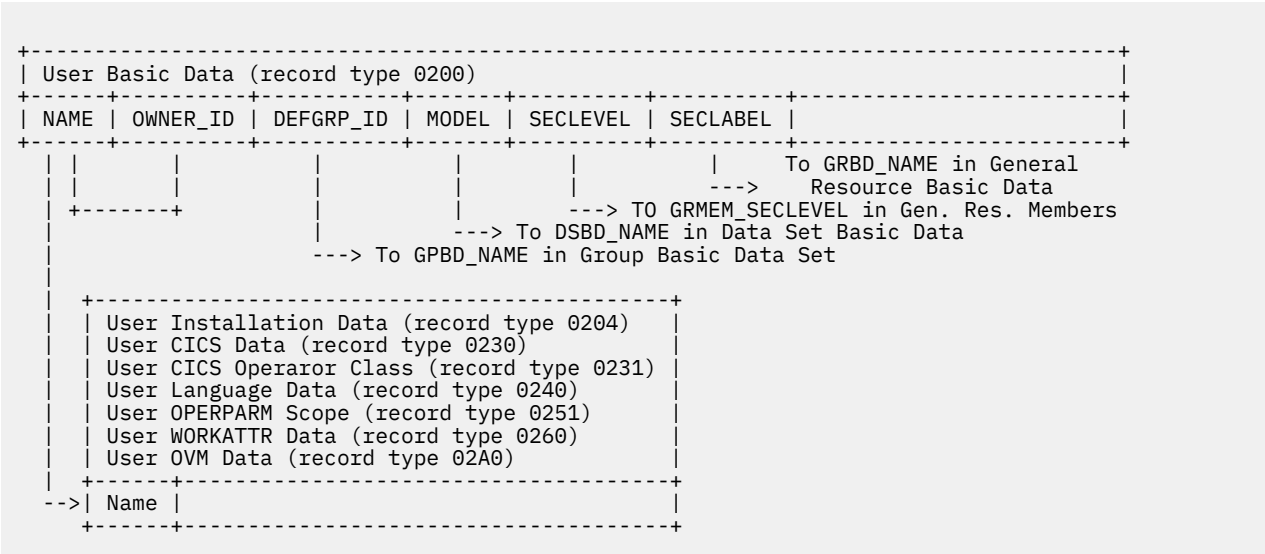


Figure 3. Relationship among the User Record Types

Figure 3 on page 268 shows the following records and their relationships:

User Basic Data (record type 0200)
Has the following fields and pointers:

- NAME
- OWNER_ID, which points to NAME
- DEFGRP_ID, which point to GPBD_NAME in Group Basic Data
- MODEL, which points to DSBD_NAME in Data Set Basic Data
- SECLEVEL, which points to GRMEM_SECLEVEL in General Resource Members
- SECLABEL, which points to GRBD_NAME in General Resource Basic Data.

User Installation Data (record type 0204)
User CICS Data (record type 0230)
User CICS Operator Class (record type 0231)
User Language Data (record type 0240)
User OPERPARM Scope (record type 0251)
User WORKATTR Data (record type 0260)
User OVM Data (record type 02A0)
Each of which has a NAME field, which points to NAME in User Basic Data (record type 0200).

These additional records and their relationships are not shown in Figure 3 on page 268:

User Basic Data (record type 0200)
Has the following fields and pointers:

- NAME
- OWNER_ID, which points to NAME
- DEFGRP_ID, which points to GPBD_NAME in Group Basic Data
- MODEL, which points to DSBD_NAME in Data Set Basic Data
- SECLEVEL, which points to GRMEM_SECLEVEL in General Resource Members
- SECLABEL, which points to GRBD_NAME in General Resource Basic Data.

User Categories (record type 0201)
Has the following fields and pointers:

- NAME, which points to NAME in User Basic Data (record type 0200)
- CATEGORY, which points to GRMEM_CATEGORY in General Resource Members.

User Classes (record type 0202)
Has the following fields and pointers:

- NAME, which points to NAME in User Basic Data (record type 0200)
- CLASS_NAME, which points to CLASS_NAME in all general resource record types.

User Group Connection Data (record type 0203)
Has the following fields and pointers:

- NAME, which points to NAME in User Basic Data (record type 0200)
- GRP_ID, which points to GPBD_NAME in Group Basic Data.

User Connect Data (record type 0205)
Has the following fields and pointers:

- NAME, which points to NAME in User Basic Data (record type 0200)
- GRP_ID, which points to GPBD_NAME in Group Basic Data
- OWNER_ID, which points to DEFGRP_ID in User Basic Data (record type 0200) and to GPBD_NAME in Group Basic Data.

User DFP Data (record type 0210)
Has the following fields and pointers:

- NAME, which points to NAME in User Basic Data (record type 0200)
- MGMTCLAS, which points to GRBD_NAME in General Resource Basic Data
- STORCLAS, which points to GRBD_NAME in General Resource Basic Data.

User TSO Data (record type 0220)
Has the following fields and pointers:

- NAME, which points to NAME in User Basic Data (record type 0200)
- ACCOUNT, which points to GRBD_NAME in General Resource Basic Data
- SECLABEL, which points to GRBD_NAME in General Resource Basic Data.

User OPERPARM Data (record type 0250)
Has the following fields and pointers:

- NAME, which points to NAME in User Basic Data (record type 0200)
- ALTGRP_ID, which points to GPBD_NAME in Group Basic Data.

Data Set Records

The high level qualifier which represents the table identifier is omitted. For data set records, these qualifiers are:

Record Name	Record Type	Record Prefix
Data Set Basic Data	0400	DSBD
Data Set Categories	0401	DSCAT
Data Set Conditional Access	0402	DSCACC
Data Set Volumes	0403	DSVOL
Data Set Access	0404	DSACC
Data Set Installation Data	0405	DSINSTD

Record Name	Record Type	Record Prefix
Data Set DFP Data	0410	DSDFP

The NAME/VOL field is a concatenation of the NAME field and VOLUME field.

The following information represents data set records and their relationships:

Data Set Basic Data (record type 0400)
Has the following fields and pointers:

- NAME/VOL
- OWNER_ID, which points to USBD_NAME in User Basic Data and GPBD_NAME in Group Basic Data
- GRP_ID, which points to GPBD_NAME in Group Basic Data
- SECLEVEL, which points to GRMEM_SECLEVEL in General Resource Members
- NOTIFY_ID, which points to USBD_NAME in User Basic Data
- SECLABEL, which points to GRBD_NAME in General Resource Basic Data.

Data Set Categories (record type 0401)
Has the following fields and pointers:

- NAME/VOL, which points to NAME/VOL in Data Set Basic Data (record type 0400)
- CATEGORY, which points to GRMEM_CATEGORY in General Resource Members.

Data Set Conditional Access (record type 0402)
Has the following fields and pointers:

- NAME/VOL, which points to NAME/VOL in Data Set Basic Data (record type 0400)
- CANAME, which points to GRBD_NAME in General Resource Basic Data
- AUTH_ID, which points to GPBD_NAME in Group Basic Data and USBD_NAME in User Basic Data.

Data Set Volumes (record type 0403)
Has the following fields and pointers:

- NAME/VOL, which points to NAME/VOL in Data Set Basic Data (record type 0400)
- VOL_NAME, which points to GRBD_NAME in General Resource Basic Data.

Data Set Access (record type 0404)
Has the following fields and pointers:

- NAME/VOL, which points to NAME/VOL in Data Set Basic Data (record type 0400)
- AUTH_ID, which points to USBD_NAME in User Basic Data.

Data Set Installation Data (record type 0405)
Has the following field and pointer:

- NAME/VOL, which points to NAME/VOL in Data Set Basic Data (record type 0400).

Data Set DFP Data (record type 0410)
Has the following fields and pointers:

- NAME/VOL, which points to NAME/VOL in Data Set Basic Data (record type 0400)
- RESOWNER_ID, which points to USBD_NAME in User Basic Data.

General Resource Records

The high level qualifier which represents the table identifier is omitted. For general resource records, these qualifiers are:

Record Name	Record Type	Record Prefix
General Resource Basic Data	0500	GRBD

Record Name	Record Type	Record Prefix
General Resource Tape Volume Data	0501	GRTVOL
General Resource Categories	0502	GRCAT
General Resource Members	0503	GRMEM
General Resource Volumes	0504	GRVOL
General Resource Access	0505	GRACC
General Resource Installation Data	0506	GRINSTD
General Resource Conditional Access	0507	GRCACC
General Resource Session Data	0510	GRSES
General Resource Session Entities	0511	GRSESE
General Resource DLF Data	0520	GRDLF
General Resource DLF Job Names	0521	GRDLFJ

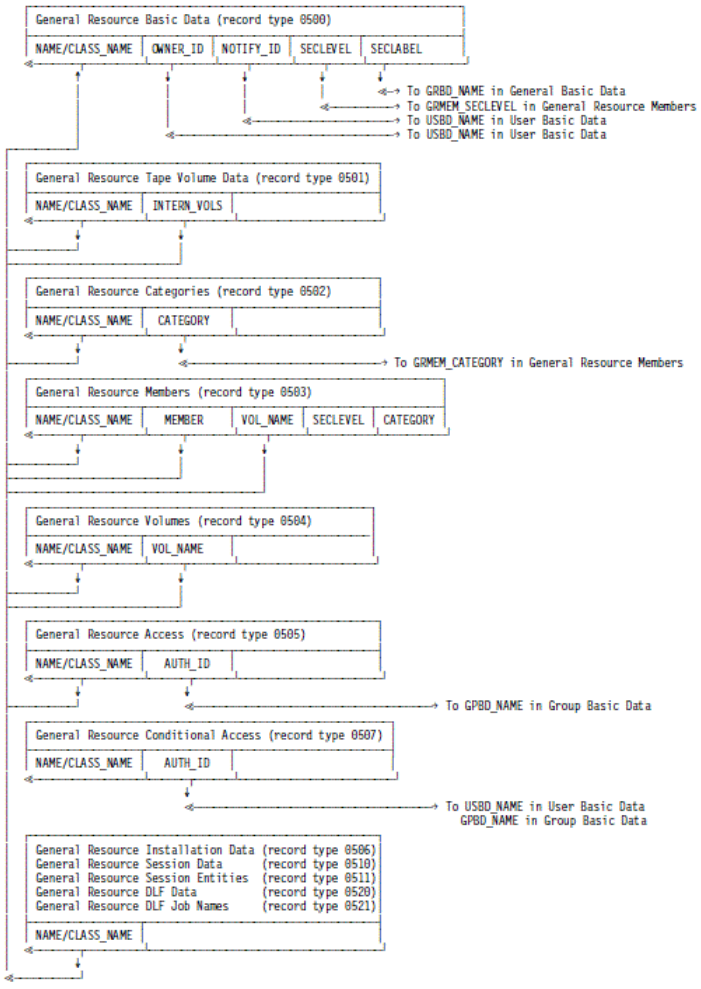


Figure 4. Relationship among the General Resource Record Types

Figure 4 on page 271 shows the following records and their relationships:

General Resource Basic Data (record type 0500)
Has the following fields and pointers:

- NAME/CLASS_NAME
- OWNER_ID, which points to USBD_NAME in User Basic Data
- NOTIFY_ID, which points to USBD_NAME in User Basic Data
- SECLEVEL, which points to GRMEM_SECLEVEL in General Resource Members
- SECLABEL, which points to GRBD_NAME in General Basic Data.

General Resource Tape Volume Data (record type 0501)
Has the following fields and pointers:

- NAME/CLASS_NAME, which points to NAME/CLASS_NAME in General Resource Basic Data (record type 0500)
- INTERN_VOLS, which points to NAME/CLASS_NAME in General Resource Basic Data (record type 0500).

General Resource Categories (record type 0502)
Has the following fields and pointers:

- NAME/CLASS_NAME, which points to NAME/CLASS_NAME in General Resource Basic Data (record type 0500)
- CATEGORY, which points to GRMEM_CATEGORY in General Resource Members.

General Resource Members (record type 0503)

- NAME/CLASS_NAME, which points to NAME/CLASS_NAME in General Resource Basic Data (record type 0500)
- MEMBER, which points to NAME/CLASS_NAME in General Resource Basic Data (record type 0500)
- VOL_NAME, which points to NAME/CLASS_NAME in General Resource Basic Data (record type 0500)
- SECLEVEL
- CATEGORY.

General Resource Volumes (record type 0504)
Has the following fields and pointers:

- NAME/CLASS_NAME, which points to NAME/CLASS_NAME in General Resource Basic Data (record type 0500)
- VOL_NAME, which points to NAME/CLASS_NAME in General Resource Basic Data (record type 0500).

General Resource Access (record type 0505)
Has the following fields and pointers:

- NAME/CLASS_NAME, which points to NAME/CLASS_NAME in General Resource Basic Data (record type 0500)
- AUTH_ID, which points to GPBD_NAME in Group Basic Data.

General Resource Conditional Access (record type 0507)
Has the following fields and pointers:

- NAME/CLASS_NAME
- AUTH_ID, which points to USBD_NAME in User Basic Data and GPBD_NAME in Group Basic Data,

General Resource Installation Data (record type 0506)
General Resource Session Data (record type 0510)
General Resource Session Entities (record type 0511)
General Resource DLF Data (record type 0520)
General Resource DLF Job Names (record type 0521)
Each of which has the following field and pointer:

- NAME/CLASS_NAME, which points to NAME/CLASS_NAME in General Resource Basic Data (record type 0500).

Conversion Rules of the Database Unload Utility

In unloading the database, these rules were followed:

- Each repeat group has its own record type

  For example, the repeat group representing the access list for data sets covered by a profile is ACL2CNT (the field name in the template). There is a data set access record (type 0404) created for each entry in the access list.
- Flag fields that are not mutually exclusive values (for example, 8-bit flags where more than one bit could be on at once) are defined as separate fields.

  When this field is processed, it is unloaded as a 4-character field, with the values YES and NO as valid values. The field is left-justified.
- Flag fields that have mutually exclusive settings are unloaded as 8-character fields with a value corresponding to each bit setting.

  For example, the UACC in a data set profile is a flag field in which each bit position corresponds to a universal access. The utility translates this single flag field into an 8-byte string with the value NONE, READ, UPDATE, CONTROL, or ALTER. If the flag field contains a value which is undefined, then the utility unloads the value as X<cc>, where cc is the hexadecimal value of the flag field.
- Encrypted and reserved fields are not unloaded.
- A maximum of 255 bytes are unloaded for any field with the exception of the HOME, PROGRAM, and FSROOT fields in a user's OVM segment for which 1023 bytes are unloaded.
- Fields for the customer's data, such as INSTDATA or the USRxx fields, are unloaded without any decoding. The USRFLG field, however, is treated as a hexadecimal value and is represented by X<cc>.
- A single byte with the value blank (X'40') is placed between each field in the output record. This makes it easier to understand the output file when it is viewed.
- Fields in the database which contain null data have blanks unloaded, with the exception of integer fields, which have a zero value unloaded.

Record Formats Produced by the Database Unload Utility

The following sections contain a detailed description of the records that are produced by the database unload utility.

Each row in the tabular description of the records that are produced by the utility contains five pieces of information:

1. Descriptive name for the field
2. Type of field

   Char
   Character data.

   Int
   Integer-EBCDIC numeric data.

   Time
   A time value, in the form hh:mm:ss.

Date
A date value, in the form yyyy-mm-dd.

Yes/No
Flag data, having the value YES or NO.

3. Starting position for the field
4. Ending position for the field
5. Free form description of the field, which may contain the valid value constraints.

The complete record formats are located as follows:

- Group records, see "Group Record Formats" on page 274
- User records, see "User Record Formats" on page 276
- Data Set records, see "Data Set Record Formats" on page 289
- General Resource records, see "General Resource Record Formats" on page 293

Group Record Formats

The records associated with groups are:

- Group Basic Data
- Group Subgroups
- Group Members
- Group Installation Data
- Group DFP Data
- Group OVM Data

Group Basic Data Record

The Group Basic Data record defines the basic information that defines a group.

There is one record per group.

Note: For some applications, such as SQL/VM, the start and end positions must account for a 4 position length indicator at the front of each record. For applications such as these, 4 would be added to each of the start and end positions indicated.

Table 145. Group Basic Data Record

Field Name	Type	Position		Comments
		Start	End	
GPBD_RECORD_TYPE	Int	1	4	Record type of the Group Basic Data record (0100).
GPBD_NAME	Char	6	13	Group name as taken from the profile name.
GPBD_SUPGRP_ID	Char	15	22	Name of the superior group to this group.
GPBD_CREATE_DATE	Date	24	33	Date that the group was defined.
GPBD_OWNER_ID	Char	35	42	The user ID or group ID which owns the profile.
GPBD_UACC	Char	44	51	The default universal access. Valid values are NONE for all groups other than the VSAMDSET group which has CREATE.
GPBD_NOTERMUACC	Yes/ No	53	56	Indicates if the group must be specifically PERMITted to use a particular terminal.
GPBD_INSTALL_DATA	Char	58	312	Installation defined data.
GPBD_MODEL	Char	314	357	Data set profile that is used as a model for this group.

Group Subgroups Record

The Group Subgroups record defines the relationship between a group and any subgroups that are within the group.

There is one record per group/subgroup combination.

Note: For some applications, such as SQL/VM, the start and end positions must account for a 4 position length indicator at the front of each record. For applications such as these, 4 would be added to each of the start and end positions indicated.

Table 146. Group Subgroups Record

Field Name	Type	Position		Comments
		Start	End	
GPSGRP_RECORD_TYPE	Int	1	4	Record type of the Group Subgroups record (0101).
GPSGRP_NAME	Char	6	13	Group name as taken from the profile name.
GPSGRP_SUBGRP_ID	Char	15	22	The name of a subgroup within the group.

Group Members Record

The Group Members record defines the relationship between a group and the members of the group.

There is one record per group/member combination.

Note: For some applications, such as SQL/VM, the start and end positions must account for a 4 position length indicator at the front of each record. For applications such as these, 4 would be added to each of the start and end positions indicated.

Table 147. Group Members Record

Field Name	Type	Position		Comments
		Start	End	
GPMEM_RECORD_TYPE	Int	1	4	Record type of the Group Members record (0102).
GPMEM_NAME	Char	6	13	Group name as taken from the profile name.
GPMEM_MEMBER_ID	Char	15	22	A user ID within the group.
GPMEM_AUTH	Char	24	31	Indicates the authority that the user ID has within the group. Valid values are USE, CONNECT, JOIN, and CREATE.

Group Installation Data Record

The Group Installation Data record defines the user data associated with a group.

This record type contains the data stored in the USRCNT repeat group, which is a field in the RACF database that is reserved for your installation's use. None of the RACF commands manipulate this field. Do not confuse this field with the GPBD_INSTALL_DATA field, shown in Table 147 on page 275, which you enter into the database using the ADDGROUP and ALTGROUP commands.

There is one record per group/installation data combination.

Note: For some applications, such as SQL/VM, the start and end positions must account for a 4 position length indicator at the front of each record. For applications such as these, 4 would be added to each of the start and end positions indicated.

Table 148. Group Installation Data Record

Field Name	Type	Position		Comments
		Start	End	
GPINSTD_RECORD_TYPE	Int	1	4	Record type of the Group Installation Data record (0103).
GPINSTD_NAME	Char	6	13	Group name as taken from the profile name.

Table 148. Group Installation Data Record (continued)

Field Name	Type	Position		Comments
		Start	End	
GPINSTD_USR_NAME	Char	15	22	The name of the installation defined field.
GPINSTD_USR_DATA	Char	24	278	The data for the installation defined field.
GPINSTD_USR_FLAG	Char	280	287	The flag for the installation defined field in the form X<cc>.

Group DFP Data Record

The Group DFP Data record defines the information required by the System Managed Storage (SMS) facility of the Data Facility Product (DFP). The fields in these records define the characteristics of the data that this profile protects.

There is one record per group/DFP data combination.

Note: For some applications, such as SQL/VM, the start and end positions must account for a 4 position length indicator at the front of each record. For applications such as these, 4 would be added to each of the start and end positions indicated.

Table 149. Group DFP Data Record

Field Name	Type	Position		Comments
		Start	End	
GPDFP_RECORD_TYPE	Int	1	4	Record type of the Group DFP Data record (0110).
GPDFP_NAME	Char	6	13	Group name as taken from the profile name.
GPDFP_DATAAPPL	Char	15	22	Default application name for the group.
GPDFP_DATACLAS	Char	24	31	Default data class for the group.
GPDFP_MGMTCLAS	Char	33	40	Default management class for the group.
GPDFP_STORCLAS	Char	42	49	Default storage class for the group.

Group OVM Data Record

The Group OVM Data record defines the OpenExtensions GIDs that have been assigned to RACF groups.

There is one record per group/GID data combination.

Note: For some applications, such as SQL/VM, the start and end positions must account for a 4 position length indicator at the front of each record. For applications such as these, 4 would be added to each of the start and end positions indicated.

Table 150. Group OVM Data Record

Field Name	Type	Position		Comments
		Start	End	
GPOVM_RECORD_TYPE	Int	1	4	Record type of the Group OVM Data record (0130).
GPOVM_NAME	Char	6	13	Group name as taken from the profile name.
GPOVM_GID	Char	15	24	OVM GID associated with the group name from the profile.

User Record Formats

The records associated with users are:

- User Basic Data
- User Categories
- User Classes

- User Group Connections
- User Installation Data
- User Connect Data
- User DFP Data
- User TSO Data
- User CICS Data
- User CICS Operator Classes
- User Language Data
- User OPERPARM Data
- User OPERPARM Scope
- User WORKATTR Data
- User OVM Data

User Basic Data Record

The User Basic Data record defines the basic information about a user.

There is one record per user.

Note: For some applications, such as SQL/VM, the start and end positions must account for a 4 position length indicator at the front of each record. For applications such as these, 4 would be added to each of the start and end positions indicated.

Table 151. User Basic Data Record

Field Name	Type	Position		Comments
		Start	End	
USBD_RECORD_TYPE	Int	1	4	Record type of the User Basic Data record (0200).
USBD_NAME	Char	6	13	User ID as taken from the profile name.
USBD_CREATE_DATE	Date	15	24	The date that the profile was created.
USBD_OWNER_ID	Char	26	33	The user ID or group ID which owns the profile.
USBD_ADSP	Yes/No	35	38	Does the user have the ADSP attribute?
USBD_SPECIAL	Yes/No	40	43	Does the user have the SPECIAL attribute?
USBD_OPER	Yes/No	45	48	Does the user have the OPERATIONS attribute?
USBD_REVOKE	Yes/No	50	53	Is the user REVOKEd?
USBD_GRPACC	Yes/No	55	58	Does the user have the GRPACC attribute?
USBD_PWD_INTERVAL	Int	60	62	The number of days that the user's password may be used.
USBD_PWD_DATE	Date	64	73	The date that the password was last changed.
USBD_PROGRAMMER	Char	75	94	The name associated with the user ID.
USBD_DEFGRP_ID	Char	96	103	The default group associated with the user.
USBD_LASTJOB_TIME	Time	105	112	The time that the user last entered the system.
USBD_LASTJOB_DATE	Date	114	123	The date that the user last entered the system.
USBD_INSTALL_DATA	Char	125	379	Installation defined data.
USBD_UAUDIT	Yes/No	381	384	Do all RACHECK and RACDEF SVCs cause logging?
USBD_AUDITOR	Yes/No	386	389	Does the user have the AUDITOR attribute?
USBD_PROTECTED	Yes/No	391	394	Does the user have the PROTECTED attribute?
USBD_OIDCARD	Yes/No	396	399	Does the user have OIDCARD data?
USBD_MFA	Yes/No	401	404	Does the user have the MFA attribute?

Table 151. User Basic Data Record (continued)

Field Name	Type	Position		Comments
		Start	End	
USBD_PWFALLBACK	Yes/No	406	409	Does the user have the PWFALLBACK attribute?
USBD_PWD_GEN	Int	411	413	The current password generation number.
USBD_REVOKE_CNT	Int	415	417	The number of unsuccessful logon attempts.
USBD_MODEL	Char	419	462	The data set model profile name.
USBD_SECLEVEL	Int	464	466	The user's security level.
USBD_REVOKE_DATE	Date	468	477	The date that the user will be revoked.
USBD_RESUME_DATE	Date	479	488	The date that the user will be resumed.
USBD_ACCESS_SUN	Yes/No	490	493	Can the user access the system on Sunday?
USBD_ACCESS_MON	Yes/No	495	498	Can the user access the system on Monday?
USBD_ACCESS_TUE	Yes/No	500	503	Can the user access the system on Tuesday?
USBD_ACCESS_WED	Yes/No	505	508	Can the user access the system on Wednesday?
USBD_ACCESS_THU	Yes/No	510	513	Can the user access the system on Thursday?
USBD_ACCESS_FRI	Yes/No	515	518	Can the user access the system on Friday?
USBD_ACCESS_SAT	Yes/No	520	523	Can the user access the system on Saturday?
USBD_START_TIME	Time	525	532	After what time may the user logon?
USBD_END_TIME	Time	534	541	After what time may the user not logon?
USBD_SECLABEL	Char	543	550	The user's default security label.
USBD_ATTRIBS	Char	552	559	Other user attributes (RSTD for users with RESTRICTED attribute).
USBD_PWDENV_EXISTS	Yes/ No	561	564	Has a PKCS#7 envelope been created for the user's current password?
USBD_PWD_ASIS	Yes/ No	566	569	Should the password be evaluated in the case entered?
USBD_PHR_DATE	Date	571	580	The date the password phrase was last changed.
USBD_PHR_GEN	Int	582	584	The current password phrase generation number.
USBD_CERT_SEQN	Int	586	595	Sequence number that is incremented whenever a certificate for the user is added, deleted, or altered.
USBD_PPHENV_EXISTS	Yes/No	597	600	Has the user's current pass phrase been PKCS#7 enveloped for possible retrieval?
USBD_PWD_ALG	Char	602	613	Algorithm that is used to protect passwords. Possible values are "LEGACY", "KDFAES". and "NOPASSWORD".
USBD_LEG_PWDHIST_CT	Int	615	617	Number of legacy password history entries.
USBD_XPW_PWDHIST_CT	Int	619	621	Number of KDFAES password history entries.
USBD_PHR_ALG	Char	623	634	Algorithm that is used to protect password phrases. Possible values are "LEGACY", "KDFAES". and "NOPHRASE".
USBD_LEG_PHRHIST_CT	Int	636	638	Number of legacy password phrase history entries.
USBD_XPW_PHRHIST_CT	Int	640	642	Number of KDFAES password phrase history entries.
USBD_ROAUDIT	Yes/No	644	647	Does the user have the ROAUDIT attribute?

User Categories Record

The User Categories record defines the categories to which the user has access.

There is one record per user/category combination.

Note: For some applications, such as SQL/VM, the start and end positions must account for a 4 position length indicator at the front of each record. For applications such as these, 4 would be added to each of the start and end positions indicated.

Table 152. User Categories Record

Field Name	Type	Position		Comments
		Start	End	
USCAT_RECORD_TYPE	Int	1	4	Record type of the User Categories record (0201).
USCAT_NAME	Char	6	13	User ID as taken from the profile name.
USCAT_CATEGORY	Int	15	19	Category to which the user has access.

User Classes Record

The User Classes record defines the classes in which the user can create profiles.

There is one record per user/class combination.

Note: For some applications, such as SQL/VM, the start and end positions must account for a 4 position length indicator at the front of each record. For applications such as these, 4 would be added to each of the start and end positions indicated.

Table 153. User Classes Record

Field Name	Type	Position		Comments
		Start	End	
USCLA_RECORD_TYPE	Int	1	4	Record type of the User Classes record (0202).
USCLA_NAME	Char	6	13	User ID as taken from the profile name.
USCLA_CLASS	Char	15	22	A class in which the user is allowed to define profiles.

User Group Connections Record

The User Group Connections record defines the groups with which the user is associated.

There is one record per user connection.

Note: For some applications, such as SQL/VM, the start and end positions must account for a 4 position length indicator at the front of each record. For applications such as these, 4 would be added to each of the start and end positions indicated.

Table 154. User Group Connections Record

Field Name	Type	Position		Comments
		Start	End	
USGCON_RECORD_TYPE	Int	1	4	Record type of the User Group Connections record (0203).
USGCON_NAME	Char	6	13	User ID as taken from the profile name.
USGCON_GRP_ID	Char	15	22	The group with which the user is associated.

User Installation Data Record

The User Installation Data record defines the user data associated with a user ID.

This record type contains the data stored in the USRCNT repeat group, which is a field in the RACF database that is reserved for your installation's use. None of the RACF commands manipulate this field. Do not confuse this field with the USER_INSTALL_DATA field, shown in Table 151 on page 277, which you enter into the database using the ADDUSER and ALTUSER commands.

Note: For some applications, such as SQL/VM, the start and end positions must account for a 4 position length indicator at the front of each record. For applications such as these, 4 would be added to each of the start and end positions indicated.

Table 155. User Installation Data Record				
Field Name	**Type**	**Position**		**Comments**
		Start	**End**	
USINSTD_RECORD_TYPE	Int	1	4	Record type of the User Installation Data record (0204).
USINSTD_NAME	Char	6	13	User ID as taken from the profile name.
USINSTD_USR_NAME	Char	15	22	The name of the installation defined field.
USINSTD_USR_DATA	Char	24	278	The data for the installation defined field.
USINSTD_USR_FLAG	Char	280	287	The flag for the installation defined field in the form X<cc>.

User Connect Data Record

The User Connect Data record defines the relationships between users and groups.

There is one record per user connection.

Note: For some applications, such as SQL/VM, the start and end positions must account for a 4 position length indicator at the front of each record. For applications such as these, 4 would be added to each of the start and end positions indicated.

Table 156. User Connect Data Record				
Field Name	**Type**	**Position**		**Comments**
		Start	**End**	
USCON_RECORD_TYPE	Int	1	4	Record type of the User Connect Data record (0205).
USCON_NAME	Char	6	13	User ID as taken from the profile name.
USCON_GRP_ID	Char	15	22	The group name.
USCON_CONNECT_DATE	Date	24	33	The date that the user was connected.
USCON_OWNER_ID	Char	35	42	The owner of the user-group connection.
USCON_LASTCON_TIME	Time	44	51	Time that the user last connected to this group.
USCON_LASTCON_DATE	Date	53	62	Date that the user last connected to this group.
USCON_UACC	Char	64	71	The default universal access. Valid values are NONE for all user IDs other than IBMUSER, which has READ to SYS1, SYSCATLG, and VSAMDSET.
USCON_INIT_CNT	Int	73	77	The number of RACINITs issued for this user/group combination.
USCON_GRP_ADSP	Yes/No	79	82	Does this user have the ADSP attribute in this group?
USCON_GRP_SPECIAL	Yes/No	84	87	Does this user have GROUP-SPECIAL in this group?
USCON_GRP_OPER	Yes/No	89	92	Does this user have GROUP-OPERATIONS in this group?
USCON_REVOKE	Yes/No	94	97	Is this user revoked?
USCON_GRP_ACC	Yes/No	99	102	Does this user have the GRPACC attribute?
USCON_NOTERMUACC	Yes/No	104	107	Does this user have the NOTERMUACC attribute in this group?
USCON_GRP_AUDIT	Yes/No	109	112	Does this user have the GROUP-AUDITOR attribute in this group?
USCON_REVOKE_DATE	Date	114	123	The date that the user's connection to the group will be revoked.
USCON_RESUME_DATE	Date	125	134	The date that the user's connection to the group will be resumed.

User DFP Data Record

The User DFP Data record defines the information required by the System Managed Storage facility of the Data Facility Product (DFP). The fields in these records define the characteristics of the data that are created by the user.

There is one record per user/DFP data combination.

Note: For some applications, such as SQL/VM, the start and end positions must account for a 4 position length indicator at the front of each record. For applications such as these, 4 would be added to each of the start and end positions indicated.

Table 157. User DFP Data Record

Field Name	Type	Position		Comments
		Start	End	
USDFP_RECORD_TYPE	Int	1	4	Record type of the User DFP data record (0210).
USDFP_NAME	Char	6	13	User ID as taken from the profile name.
USDFP_DATAAPPL	Char	15	22	Default application name for the user.
USDFP_DATACLAS	Char	24	31	Default data class for the user.
USDFP_MGMTCLAS	Char	33	40	Default management class for the user.
USDFP_STORCLAS	Char	42	49	Default storage class for the user.

User TSO Data Record

The User TSO Data record defines the information required by the MVS™ Time Sharing Option (TSO) interactive environment.

There is one record per TSO user.

Note: For some applications, such as SQL/VM, the start and end positions must account for a 4 position length indicator at the front of each record. For applications such as these, 4 would be added to each of the start and end positions indicated.

Table 158. User TSO Data Record

Field Name	Type	Position		Comments
		Start	End	
USTSO_RECORD_TYPE	Int	1	4	Record type of the User TSO Data record (0220).
USTSO_NAME	Char	6	13	User ID as taken from the profile name.
USTSO_ACCOUNT	Char	15	54	The default account number.
USTSO_COMMAND	Char	56	135	The command issued at LOGON.
USTSO_DEST	Char	137	144	The default destination identifier.
USTSO_HOLD_CLASS	Char	146	146	The default hold class.
USTSO_JOB_CLASS	Char	148	148	The default job class.
USTSO_LOGON_PROC	Char	150	157	The default logon procedure.
USTSO_LOGON_SIZE	Int	159	168	The default logon region size.
USTSO_MSG_CLASS	Char	170	170	The default message class.
USTSO_LOGON_MAX	Int	172	181	The maximum logon region size.
USTSO_PERF_GROUP	Int	183	192	The performance group associated with the user.
USTSO_SYSOUT_CLASS	Char	194	194	The default sysout class.
USTSO_USER_DATA	Char	196	203	The TSO user data, in hexadecimal in the form X<cccc>.
USTSO_UNIT_NAME	Char	205	212	The default SYSDA device.

Table 158. User TSO Data Record (continued)

Field Name	Type	Position		Comments
		Start	End	
USTSO_SECLABEL	Char	214	221	The default logon security label.

User CICS Data Record

The User CICS Data record defines the data required by the Customer Information Control System (CICS).

There is one record per user/CICS data combination.

Note: For some applications, such as SQL/VM, the start and end positions must account for a 4 position length indicator at the front of each record. For applications such as these, 4 would be added to each of the start and end positions indicated.

Table 159. User CICS Data Record

Field Name	Type	Position		Comments
		Start	End	
USCICS_RECORD_TYPE	Int	1	4	Record type of the User CICS Data record (0230).
USCICS_NAME	Char	6	13	User ID as taken from the profile name.
USCICS_OPIDENT	Char	15	17	The CICS operator identifier.
USCICS_OPPRTY	Int	19	23	The CICS operator priority.
USCICS_NOFORCE	Yes/No	25	28	Is the extended recovery facility (XRF) NOFORCE option in effect?
USCICS_TIMEOUT	Int	30	34	The terminal time-out value.

User CICS Operator Classes Record

The User CICS Operator Classes record defines the classes associated with a CICS operator.

There is one record per user/CICS operator class combination.

Note: For some applications, such as SQL/VM, the start and end positions must account for a 4 position length indicator at the front of each record. For applications such as these, 4 would be added to each of the start and end positions indicated.

Table 160. User CICS Operator Class Record

Field Name	Type	Position		Comments
		Start	End	
USCOPC_RECORD_TYPE	Int	1	4	Record type of the User CICS Operator Class record (0231).
USCOPC_NAME	Char	6	13	User ID as taken from the profile name.
USCOPC_OPCLASS	Char	15	17	The class associated with the CICS operator.

User Language Data Record

The User Language Data record defines the primary and default languages for the user.

There is one record per user/language combination.

Note: For some applications, such as SQL/VM, the start and end positions must account for a 4 position length indicator at the front of each record. For applications such as these, 4 would be added to each of the start and end positions indicated.

Table 161. User Language Data Record				
Field Name	**Type**	**Position**		**Comments**
		Start	**End**	
USLAN_RECORD_TYPE	Int	1	4	Record type of the User Language Data record (0240).
USLAN_NAME	Char	6	13	User ID as taken from the profile name.
USLAN_PRIMARY	Char	15	17	The primary language for the user.
USLAN_SECONDARY	Char	19	21	The secondary language for the user.

User OPERPARM Data Record

The User OPERPARM Data record defines the operator characteristics for the user.

There is one record per user/OPERPARM data combination.

Note: For some applications, such as SQL/VM, the start and end positions must account for a 4 position length indicator at the front of each record. For applications such as these, 4 would be added to each of the start and end positions indicated.

Table 162. User OPERPARM Data Record				
Field Name	**Type**	**Position**		**Comments**
		Start	**End**	
USOPR_RECORD_TYPE	Int	1	4	Record type of the User OPERPARM Data record (0250).
USOPR_NAME	Char	6	13	User ID as taken from the profile name.
USOPR_STORAGE	Int	15	19	The number of megabytes of storage that can be used for message queuing.
USOPR_MASTERAUTH	Yes/No	21	24	Does this user have MASTER console authority?
USOPR_ALLAUTH	Yes/No	26	29	Does this user have ALL console authority?
USOPR_SYSAUTH	Yes/No	31	34	Does this user have SYSAUTH console authority?
USOPR_IOAUTH	Yes/No	36	39	Does this user have I/O console authority?
USOPR_CONSAUTH	Yes/No	41	44	Does this user have CONS console authority?
USOPR_INFOAUTH	Yes/No	46	49	Does this user have INFO console authority?
USOPR_TIMESTAMP	Yes/No	51	54	Do console messages contain a timestamp?
USOPR_SYSTEMID	Yes/No	56	59	Do console messages contain a system ID?
USOPR_JOBID	Yes/No	61	64	Do console messages contain a job ID?
USOPR_MSGID	Yes/No	66	69	Do console messages contain a message ID?
USOPR_X	Yes/No	71	74	Are the job name and system name to be suppressed for messages issued from the JES3 global processor?
USOPR_WTOR	Yes/No	76	79	Does the console receive WTOR messages?
USOPR_IMMEDIATE	Yes/No	81	84	Does the console receive *immediate* messages?
USOPR_CRITICAL	Yes/No	86	89	Does the console receive *critical event* messages?
USOPR_EVENTUAL	Yes/No	91	94	Does the console receive *eventual event* messages?
USOPR_INFO	Yes/No	96	99	Does the console receive *informational* messages?
USOPR_NOBRODCAST	Yes/No	101	104	Are broadcast messages to this console suppressed?
USOPR_ALL	Yes/No	106	109	Does the console receive `all` messages?
USOPR_JOBNAMES	Yes/No	111	114	Are job names monitored?
USOPR_JOBNAMEST	Yes/No	116	119	Are job names monitored with timestamps displayed?

Table 162. User OPERPARM Data Record (continued)

Field Name	Type	Position		Comments
		Start	End	
USOPR_SESS	Yes/No	121	124	Are user IDs displayed with each TSO initiation and termination?
USOPR_SESST	Yes/No	126	129	Are user IDs and timestamps displayed with each TSO initiation and termination?
USOPR_STATUS	Yes/No	131	134	Are data set names and dispositions displayed with each data set that is freed?
USOPR_ROUTECODE001	Yes/No	136	139	Is this console enabled for route code 001?
USOPR_ROUTECODE002	Yes/No	141	144	Is this console enabled for route code 002?
USOPR_ROUTECODE003	Yes/No	146	149	Is this console enabled for route code 003?
USOPR_ROUTECODE004	Yes/No	151	154	Is this console enabled for route code 004?
USOPR_ROUTECODE005	Yes/No	156	159	Is this console enabled for route code 005?
USOPR_ROUTECODE006	Yes/No	161	164	Is this console enabled for route code 006?
USOPR_ROUTECODE007	Yes/No	166	169	Is this console enabled for route code 007?
USOPR_ROUTECODE008	Yes/No	171	174	Is this console enabled for route code 008?
USOPR_ROUTECODE009	Yes/No	176	179	Is this console enabled for route code 009?
USOPR_ROUTECODE010	Yes/No	181	184	Is this console enabled for route code 010?
USOPR_ROUTECODE011	Yes/No	186	189	Is this console enabled for route code 011?
USOPR_ROUTECODE012	Yes/No	191	194	Is this console enabled for route code 012?
USOPR_ROUTECODE013	Yes/No	196	199	Is this console enabled for route code 013?
USOPR_ROUTECODE014	Yes/No	201	204	Is this console enabled for route code 014?
USOPR_ROUTECODE015	Yes/No	206	209	Is this console enabled for route code 015?
USOPR_ROUTECODE016	Yes/No	211	214	Is this console enabled for route code 016?
USOPR_ROUTECODE017	Yes/No	216	219	Is this console enabled for route code 017?
USOPR_ROUTECODE018	Yes/No	221	224	Is this console enabled for route code 018?
USOPR_ROUTECODE019	Yes/No	226	229	Is this console enabled for route code 019?
USOPR_ROUTECODE020	Yes/No	231	234	Is this console enabled for route code 020?
USOPR_ROUTECODE021	Yes/No	236	239	Is this console enabled for route code 021?
USOPR_ROUTECODE022	Yes/No	241	244	Is this console enabled for route code 022?
USOPR_ROUTECODE023	Yes/No	246	249	Is this console enabled for route code 023?
USOPR_ROUTECODE024	Yes/No	251	254	Is this console enabled for route code 024?
USOPR_ROUTECODE025	Yes/No	256	259	Is this console enabled for route code 025?
USOPR_ROUTECODE026	Yes/No	261	264	Is this console enabled for route code 026?
USOPR_ROUTECODE027	Yes/No	266	269	Is this console enabled for route code 027?
USOPR_ROUTECODE028	Yes/No	271	274	Is this console enabled for route code 028?
USOPR_ROUTECODE029	Yes/No	276	279	Is this console enabled for route code 029?
USOPR_ROUTECODE030	Yes/No	281	284	Is this console enabled for route code 030?
USOPR_ROUTECODE031	Yes/No	286	289	Is this console enabled for route code 031?
USOPR_ROUTECODE032	Yes/No	291	294	Is this console enabled for route code 032?
USOPR_ROUTECODE033	Yes/No	296	299	Is this console enabled for route code 033?
USOPR_ROUTECODE034	Yes/No	301	304	Is this console enabled for route code 034?

Table 162. User OPERPARM Data Record (continued)

Field Name	Type	Position		Comments
		Start	End	
USOPR_ROUTECODE035	Yes/No	306	309	Is this console enabled for route code 035?
USOPR_ROUTECODE036	Yes/No	311	314	Is this console enabled for route code 036?
USOPR_ROUTECODE037	Yes/No	316	319	Is this console enabled for route code 037?
USOPR_ROUTECODE038	Yes/No	321	324	Is this console enabled for route code 038?
USOPR_ROUTECODE039	Yes/No	326	329	Is this console enabled for route code 039?
USOPR_ROUTECODE040	Yes/No	331	334	Is this console enabled for route code 040?
USOPR_ROUTECODE041	Yes/No	336	339	Is this console enabled for route code 041?
USOPR_ROUTECODE042	Yes/No	341	344	Is this console enabled for route code 042?
USOPR_ROUTECODE043	Yes/No	346	349	Is this console enabled for route code 043?
USOPR_ROUTECODE044	Yes/No	351	354	Is this console enabled for route code 044?
USOPR_ROUTECODE045	Yes/No	356	359	Is this console enabled for route code 045?
USOPR_ROUTECODE046	Yes/No	361	364	Is this console enabled for route code 046?
USOPR_ROUTECODE047	Yes/No	366	369	Is this console enabled for route code 047?
USOPR_ROUTECODE048	Yes/No	371	374	Is this console enabled for route code 048?
USOPR_ROUTECODE049	Yes/No	376	379	Is this console enabled for route code 049?
USOPR_ROUTECODE050	Yes/No	381	384	Is this console enabled for route code 050?
USOPR_ROUTECODE051	Yes/No	386	389	Is this console enabled for route code 051?
USOPR_ROUTECODE052	Yes/No	391	394	Is this console enabled for route code 052?
USOPR_ROUTECODE053	Yes/No	396	399	Is this console enabled for route code 053?
USOPR_ROUTECODE054	Yes/No	401	404	Is this console enabled for route code 054?
USOPR_ROUTECODE055	Yes/No	406	409	Is this console enabled for route code 055?
USOPR_ROUTECODE056	Yes/No	411	414	Is this console enabled for route code 056?
USOPR_ROUTECODE057	Yes/No	416	419	Is this console enabled for route code 057?
USOPR_ROUTECODE058	Yes/No	421	424	Is this console enabled for route code 058?
USOPR_ROUTECODE059	Yes/No	426	429	Is this console enabled for route code 059?
USOPR_ROUTECODE060	Yes/No	431	434	Is this console enabled for route code 060?
USOPR_ROUTECODE061	Yes/No	436	439	Is this console enabled for route code 061?
USOPR_ROUTECODE062	Yes/No	441	444	Is this console enabled for route code 062?
USOPR_ROUTECODE063	Yes/No	446	449	Is this console enabled for route code 063?
USOPR_ROUTECODE064	Yes/No	451	454	Is this console enabled for route code 064?
USOPR_ROUTECODE065	Yes/No	456	459	Is this console enabled for route code 065?
USOPR_ROUTECODE066	Yes/No	461	464	Is this console enabled for route code 066?
USOPR_ROUTECODE067	Yes/No	466	469	Is this console enabled for route code 067?
USOPR_ROUTECODE068	Yes/No	471	474	Is this console enabled for route code 068?
USOPR_ROUTECODE069	Yes/No	476	479	Is this console enabled for route code 069?
USOPR_ROUTECODE070	Yes/No	481	484	Is this console enabled for route code 070?
USOPR_ROUTECODE071	Yes/No	486	489	Is this console enabled for route code 071?
USOPR_ROUTECODE072	Yes/No	491	494	Is this console enabled for route code 072?

Table 162. User OPERPARM Data Record (continued)

Field Name	Type	Position		Comments
		Start	End	
USOPR_ROUTECODE073	Yes/No	496	499	Is this console enabled for route code 073?
USOPR_ROUTECODE074	Yes/No	501	504	Is this console enabled for route code 074?
USOPR_ROUTECODE075	Yes/No	506	509	Is this console enabled for route code 075?
USOPR_ROUTECODE076	Yes/No	511	514	Is this console enabled for route code 076?
USOPR_ROUTECODE077	Yes/No	516	519	Is this console enabled for route code 077?
USOPR_ROUTECODE078	Yes/No	521	524	Is this console enabled for route code 078?
USOPR_ROUTECODE079	Yes/No	526	529	Is this console enabled for route code 079?
USOPR_ROUTECODE080	Yes/No	531	534	Is this console enabled for route code 080?
USOPR_ROUTECODE081	Yes/No	536	539	Is this console enabled for route code 081?
USOPR_ROUTECODE082	Yes/No	541	544	Is this console enabled for route code 082?
USOPR_ROUTECODE083	Yes/No	546	549	Is this console enabled for route code 083?
USOPR_ROUTECODE084	Yes/No	551	554	Is this console enabled for route code 084?
USOPR_ROUTECODE085	Yes/No	556	559	Is this console enabled for route code 085?
USOPR_ROUTECODE086	Yes/No	561	564	Is this console enabled for route code 086?
USOPR_ROUTECODE087	Yes/No	566	569	Is this console enabled for route code 087?
USOPR_ROUTECODE088	Yes/No	571	574	Is this console enabled for route code 088?
USOPR_ROUTECODE089	Yes/No	576	579	Is this console enabled for route code 089?
USOPR_ROUTECODE090	Yes/No	581	584	Is this console enabled for route code 090?
USOPR_ROUTECODE091	Yes/No	586	589	Is this console enabled for route code 091?
USOPR_ROUTECODE092	Yes/No	591	594	Is this console enabled for route code 092?
USOPR_ROUTECODE093	Yes/No	596	599	Is this console enabled for route code 093?
USOPR_ROUTECODE094	Yes/No	601	604	Is this console enabled for route code 094?
USOPR_ROUTECODE095	Yes/No	606	609	Is this console enabled for route code 095?
USOPR_ROUTECODE096	Yes/No	611	614	Is this console enabled for route code 096?
USOPR_ROUTECODE097	Yes/No	616	619	Is this console enabled for route code 097?
USOPR_ROUTECODE098	Yes/No	621	624	Is this console enabled for route code 098?
USOPR_ROUTECODE099	Yes/No	626	629	Is this console enabled for route code 099?
USOPR_ROUTECODE100	Yes/No	631	634	Is this console enabled for route code 100?
USOPR_ROUTECODE101	Yes/No	636	639	Is this console enabled for route code 101?
USOPR_ROUTECODE102	Yes/No	641	644	Is this console enabled for route code 102?
USOPR_ROUTECODE103	Yes/No	646	649	Is this console enabled for route code 103?
USOPR_ROUTECODE104	Yes/No	651	654	Is this console enabled for route code 104?
USOPR_ROUTECODE105	Yes/No	656	659	Is this console enabled for route code 105?
USOPR_ROUTECODE106	Yes/No	661	664	Is this console enabled for route code 106?
USOPR_ROUTECODE107	Yes/No	666	669	Is this console enabled for route code 107?
USOPR_ROUTECODE108	Yes/No	671	674	Is this console enabled for route code 108?
USOPR_ROUTECODE109	Yes/No	676	679	Is this console enabled for route code 109?
USOPR_ROUTECODE110	Yes/No	681	684	Is this console enabled for route code 110?

Table 162. User OPERPARM Data Record (continued)

Field Name	Type	Position		Comments
		Start	End	
USOPR_ROUTECODE111	Yes/No	686	689	Is this console enabled for route code 111?
USOPR_ROUTECODE112	Yes/No	691	694	Is this console enabled for route code 112?
USOPR_ROUTECODE113	Yes/No	696	699	Is this console enabled for route code 113?
USOPR_ROUTECODE114	Yes/No	701	704	Is this console enabled for route code 114?
USOPR_ROUTECODE115	Yes/No	706	709	Is this console enabled for route code 115?
USOPR_ROUTECODE116	Yes/No	711	714	Is this console enabled for route code 116?
USOPR_ROUTECODE117	Yes/No	716	719	Is this console enabled for route code 117?
USOPR_ROUTECODE118	Yes/No	721	724	Is this console enabled for route code 118?
USOPR_ROUTECODE119	Yes/No	726	729	Is this console enabled for route code 119?
USOPR_ROUTECODE120	Yes/No	731	734	Is this console enabled for route code 120?
USOPR_ROUTECODE121	Yes/No	736	739	Is this console enabled for route code 121?
USOPR_ROUTECODE122	Yes/No	741	744	Is this console enabled for route code 122?
USOPR_ROUTECODE123	Yes/No	746	749	Is this console enabled for route code 123?
USOPR_ROUTECODE124	Yes/No	751	754	Is this console enabled for route code 124?
USOPR_ROUTECODE125	Yes/No	756	759	Is this console enabled for route code 125?
USOPR_ROUTECODE126	Yes/No	761	764	Is this console enabled for route code 126?
USOPR_ROUTECODE127	Yes/No	766	769	Is this console enabled for route code 127?
USOPR_ROUTECODE128	Yes/No	771	774	Is this console enabled for route code 128?
USOPR_LOGCMDRESP	Char	776	783	Specifies the logging of command responses received by the extended operator. Valid values are SYSTEM, NO, and blank.
USOPR_MIGRATIONID	Yes/No	785	788	Is this extended operator to receive a migration ID?
USOPR_DELOPERMSG	Char	790	797	Does this extended operator receive delete operator messages? Valid values are NORMAL, ALL, and NONE.
USOPR_RETRIEVE_KEY	Char	799	806	Specifies a retrieval key used for searching. A null value is indicated by NONE.
USOPR_CMDSYS	Char	808	815	The name of the system that the extended operator is connected to for command processing.
USOPR_UD	Yes/No	817	820	Is this operator to receive undeliverable messages?
USOPR_ALTGRP_ID	Char	822	829	The default group associated with this operator.
USOPR_AUTO	Yes/No	831	834	Is this operator to receive messages automated within the sysplex?

User OPERPARM Scope

The User OPERPARM Scope record defines the scope of the operator.

There is one record per user/OPERPARM scope combination.

Note: For some applications, such as SQL/VM, the start and end positions must account for a 4 position length indicator at the front of each record. For applications such as these, 4 would be added to each of the start and end positions indicated.

Table 163. User OPERPARM Scope Record				
Field Name	**Type**	**Position**		**Comments**
		Start	**End**	
USOPRP_RECORD_TYPE	Int	1	4	Record type of the User OPERPARM Scope record (0251).
USOPRP_NAME	Char	6	13	User ID as taken from the profile name.
USOPRP_SYSTEM	Char	15	22	System name.

User WORKATTR Data Record

The User WORKATTR Data record defines the logistical information for the user.

There is one record per user/WORKATTR data combination.

Note: For some applications, such as SQL/VM, the start and end positions must account for a 4 position length indicator at the front of each record. For applications such as these, 4 would be added to each of the start and end positions indicated.

Table 164. User WORKATTR Data Record				
Field Name	**Type**	**Position**		**Comments**
		Start	**End**	
USWRK_RECORD_TYPE	Int	1	4	Record type of the User WORKATTR Data record (0260).
USWRK_NAME	Char	6	13	User ID as taken from the profile name.
USWRK_AREA_NAME	Char	15	74	Area for delivery.
USWRK_BUILDING	Char	76	135	Building for delivery.
USWRK_DEPARTMENT	Char	137	196	Department for delivery.
USWRK_ROOM	Char	198	257	Room for delivery.
USWRK_ADDR_LINE1	Char	259	318	Address line 1.
USWRK_ADDR_LINE2	Char	320	379	Address line 2.
USWRK_ADDR_LINE3	Char	381	440	Address line 3.
USWRK_ADDR_LINE4	Char	442	501	Address line 4.
USWRK_ACCOUNT	Char	503	757	Account number.

User OVM Data Record

The User OVM Data record defines the information required by OpenExtensions z/VM. These records define the UIDs which have been assigned to RACF users, their default directory, default program name, and the file system root.

There is only one record per user/UID data combination.

Note: For some applications, such as SQL/VM, the start and end positions must account for a 4 position length indicator at the front of each record. For applications such as these, 4 would be added to each of the start and end positions indicated.

Table 165. User OVM Data Record				
Field Name	**Type**	**Position**		**Comments**
		Start	**End**	
USOVM_RECORD_TYPE	Int	1	4	Record type of the User OVM Data record (02A0).
USOVM_NAME	Char	6	13	User name as taken from the profile name.
USOVM_UID	Char	15	24	OVM UID associated with the user name from the profile.
USOVM_HOME_PATH	Char	26	1048	OVM home path associated with the UID.

Table 165. User OVM Data Record (continued)				
Field Name	**Type**	**Position**		**Comments**
		Start	**End**	
USOVM_PROGRAM	Char	1050	2072	OVM Default Program associated with the UID.
USOVM_FSROOT	Char	2074	3096	OVM File System root for this user.

Data Set Record Formats

The records associated with data sets are:

- Data Set Basic Data
- Data Set Categories
- Data Set Conditional Access
- Data Set Volumes
- Data Set Access
- Data Set Installation Data
- Data Set DFP Data

Data Set Basic Data Record

The Data Set Basic Data record defines the basic information for a data set.

There is one record per data set profile.

Note: For some applications, such as SQL/VM, the start and end positions must account for a 4 position length indicator at the front of each record. For applications such as these, 4 would be added to each of the start and end positions indicated.

Table 166. Data Set Basic Data Record				
Field Name	**Type**	**Position**		**Comments**
		Start	**End**	
DSBD_RECORD_TYPE	Int	1	4	Record type of the Data Set Basic Data record (0400).
DSBD_NAME	Char	6	49	Data set name as taken from the profile name.
DSBD_VOL	Char	51	56	Volume upon which this data set resides. Blank if the profile is generic, and `*MODEL` if the profile is a model profile.
DSBD_GENERIC	Yes/No	58	61	Is this a generic profile?
DSBD_CREATE_DATE	Date	63	72	Date the profile was created.
DSBD_OWNER_ID	Char	74	81	The user ID or group ID which owns the profile.
DSBD_LASTREF_DATE	Date	83	92	The date that the data set was last referenced.
DSBD_LASTCHG_DATE	Date	94	103	The date that the data set was last changed.
DSBD_ALTER_CNT	Int	105	109	The number of times that the data set was accessed with ALTER authority.
DSBD_CONTROL_CNT	Int	111	115	The number of times that the data set was accessed with CONTROL authority.
DSBD_UPDATE_CNT	Int	117	121	The number of times that the data set was accessed with UPDATE authority.
DSBD_READ_CNT	Int	123	127	The number of times that the data set was accessed with READ authority.
DSBD_UACC	Char	129	136	The universal access of this data set. Valid values are `NONE`, `EXECUTE`, `READ`, `UPDATE`, `CONTROL`, and `ALTER`.
DSBD_GRPDS	Yes/No	138	141	Is this a group data set?

Table 166. Data Set Basic Data Record (continued)

Field Name	Type	Position		Comments
		Start	End	
DSBD_AUDIT_LEVEL	Char	143	150	Indicates the level of resource-owner-specified auditing that is performed. Valid values are ALL, SUCCESS, FAIL, and NONE.
DSBD_GRP_ID	Char	152	159	The connect group of the user who created this data set.
DSBD_DS_TYPE	Char	161	168	The type of the data set. Valid values are VSAM, NONVSAM, TAPE, and MODEL.
DSBD_LEVEL	Int	170	172	The level of the data set.
DSBD_DEVICE_NAME	Char	174	181	The EBCDIC name of the device type on which the data set resides.
DSBD_GAUDIT_LEVEL	Char	183	190	Indicates the level of auditor-specified auditing that is performed. Valid values are ALL, SUCCESS, FAIL, and NONE.
DSBD_INSTALL_DATA	Char	192	446	Installation defined data.
DSBD_AUDIT_OKQUAL	Char	448	455	The resource-owner-specified successful access audit qualifier. This is set to blanks if AUDIT_LEVEL is NONE. Otherwise, it is set to either READ, UPDATE, CONTROL, or ALTER.
DSBD_AUDIT_FAQUAL	Char	457	464	The resource-owner-specified failing access audit qualifier. This is set to blanks if AUDIT_LEVEL is NONE. Otherwise, it is set to either READ, UPDATE, CONTROL, or ALTER.
DSBD_GAUDIT_OKQUAL	Char	466	473	The auditor-specified successful access audit qualifier. This is set to blanks if GAUDIT_LEVEL is NONE. Otherwise, it is set to either READ, UPDATE, CONTROL, or ALTER.
DSBD_GAUDIT_FAQUAL	Char	475	482	The auditor-specified failing access audit qualifier. This is set to blanks if GAUDIT_LEVEL is NONE. Otherwise, it is set to either READ, UPDATE, CONTROL, or ALTER.
DSBD_WARNING	Yes/No	484	487	Does this data set have the WARNING attribute?
DSBD_SECLEVEL	Int	489	491	The data set security level.
DSBD_NOTIFY_ID	Char	493	500	User ID that is notified when violations occur.
DSBD_RETENTION	Int	502	506	Retention period of the data set.
DSBD_ERASE	Yes/No	508	511	For a DASD data set, is this data set scratched when the data set is deleted?
DSBD_SECLABEL	Char	513	520	Security label of the data set.

Data Set Categories Record

The Data Set Categories record defines the categories to which a data set belongs.

There is one record per data set/category combination.

Note: For some applications, such as SQL/VM, the start and end positions must account for a 4 position length indicator at the front of each record. For applications such as these, 4 would be added to each of the start and end positions indicated.

Table 167. Data Set Categories Record

Field Name	Type	Position		Comments
		Start	End	
DSCAT_RECORD_TYPE	Int	1	4	Record type of the Data Set Categories record (0401).
DSCAT_NAME	Char	6	49	Data set name as taken from the profile name.
DSCAT_VOL	Char	51	56	Volume upon which this data set resides. Blank if the profile is generic, and *MODEL if the profile is a model profile.

Table 167. Data Set Categories Record (continued)				
Field Name	**Type**	**Position**		**Comments**
		Start	**End**	
DSCAT_CATEGORY	Int	58	62	Category associated with this data set.

Data Set Conditional Access Record

The Data Set Conditional Access record defines the data sets which have conditional access permissions.

There is one record per data set/access combination.

Note: For some applications, such as SQL/VM, the start and end positions must account for a 4 position length indicator at the front of each record. For applications such as these, 4 would be added to each of the start and end positions indicated.

Table 168. Data Set Conditional Access Record				
Field Name	**Type**	**Position**		**Comments**
		Start	**End**	
DSCACC_RECORD_TYPE	Int	1	4	Record type of the Data Set Conditional Access record (0402).
DSCACC_NAME	Char	6	49	Data set name as taken from the profile name.
DSCACC_VOL	Char	51	56	Volume upon which this data set resides. Blank if the profile is generic, and `*MODEL` if the profile is a model profile.
DSCACC_CATYPE	Char	58	65	The type of conditional access checking that is being performed. Valid values are `PROGRAM`, `CONSOLE`, `TERMINAL` and `JESINPUT`.
DSCACC_CANAME	Char	67	74	The name of a conditional access element which is permitted access.
DSCACC_AUTH_ID	Char	76	83	The user ID or group ID that is authorized to the data set.
DSCACC_ACCESS	Char	85	92	The access of the conditional access element/user combination. Valid values are `NONE`, `EXECUTE`, `READ`, `UPDATE`, `CONTROL`, and `ALTER`.
DSCACC_ACCESS_CNT	Int	94	98	The number of times that the data set was accessed.

Data Set Volumes Record

The Data Set Volumes record defines the volumes upon which a data set resides.

There is one record per data set/volume combination. Records exist in this table only for discrete data set profiles.

Note: For some applications, such as SQL/VM, the start and end positions must account for a 4 position length indicator at the front of each record. For applications such as these, 4 would be added to each of the start and end positions indicated.

Table 169. Data Set Volumes Record				
Field Name	**Type**	**Position**		**Comments**
		Start	**End**	
DSVOL_RECORD_TYPE	Int	1	4	Record type of the Data Set Volumes record (0403).
DSVOL_NAME	Char	6	49	Data set name as taken from the profile name.
DSVOL_VOL	Char	51	56	Volume upon which this data set resides.
DSVOL_VOL_NAME	Char	58	63	A volume upon which the data set resides.

Data Set Access Record

The Data Set Access record defines the users or groups which are allowed to access data.

There is one record per data set/authorization combination.

Note: For some applications, such as SQL/VM, the start and end positions must account for a 4 position length indicator at the front of each record. For applications such as these, 4 would be added to each of the start and end positions indicated.

Table 170. Data Set Access Record

Field Name	Type	Position		Comments
		Start	End	
DSACC_RECORD_TYPE	Int	1	4	Record type of the Data Set Access Record (0404).
DSACC_NAME	Char	6	49	Data set name as taken from the profile name.
DSACC_VOL	Char	51	56	Volume upon which this data set resides. Blank if the profile is generic, and *MODEL if the profile is a model profile.
DSACC_AUTH_ID	Char	58	65	The user ID or group ID that is authorized to the data set.
DSACC_ACCESS	Char	67	74	The access allowed to the user. Valid values are NONE, EXECUTE, READ, UPDATE, CONTROL, and ALTER.
DSACC_ACCESS_CNT	Int	76	80	The number of times that the data set was accessed.

Data Set Installation Data Record

The Data Set Installation Data record defines the user data that is associated with a data set profile.

There is one record per data set/installation data combination.

This record type contains the data stored in the USRCNT repeat group, which is a field in the RACF database that is reserved for your installation's use. None of the RACF commands manipulate this field. Do not confuse this field with the DSBD_INSTALL_DATA field, shown in Table 166 on page 289, which you enter into the database using the ADDSD and ALTDSD commands.

Note: For some applications, such as SQL/VM, the start and end positions must account for a 4 position length indicator at the front of each record. For applications such as these, 4 would be added to each of the start and end positions indicated.

Table 171. Data Set Installation Data Record

Field Name	Type	Position		Comments
		Start	End	
DSINSTD_RECORD_TYPE	Int	1	4	Record type of the Data Set Installation Data Record (0405).
DSINSTD_NAME	Char	6	49	Data set name as taken from the profile name.
DSINSTD_VOL	Char	51	56	Volume upon which this data set resides. Blank if the profile is generic, and *MODEL if the profile is a model profile.
DSINSTD_USR_NAME	Char	58	65	The name of the installation defined field.
DSINSTD_USR_DATA	Char	67	321	The data for the installation defined field.
DSINSTD_USR_FLAG	Char	323	330	The flag for the installation defined field in the form X<cc>.

Data Set DFP Data Record

The Data Set DFP Data record defines the DFP information required by the System Managed Storage (SMS) facility of the Data Facility Product (DFP).

There is one record per data set/DFP data combination.

Note: For some applications, such as SQL/VM, the start and end positions must account for a 4 position length indicator at the front of each record. For applications such as these, 4 would be added to each of the start and end positions indicated.

Table 172. Data Set DFP Data Record				
Field Name	**Type**	**Position**		**Comments**
		Start	**End**	
DSDFP_RECORD_TYPE	Int	1	4	Record type of the Data Set DFP Data record (0410).
DSDFP_NAME	Char	6	49	Data set name as taken from the profile name.
DSDFP_VOL	Char	51	56	Volume upon which this data set resides. Blank if the profile is generic, and *MODEL if the profile is a model profile.
DSDFP_RESOWNER_ID	Char	58	65	The resource owner of the data set.

General Resource Record Formats

The records associated with general resources are:

- General Resource Basic Data
- General Resource Tape Volume Data
- General Resource Categories
- General Resource Members
- General Resource Volumes
- General Resource Access
- General Resource Installation Data
- General Resource Conditional Access Data
- General Resource Session Data
- General Resource Session Entities
- General Resource DLF Data
- General Resource DLF Job Names

General Resource Basic Data Record

The General Resource Basic Data record defines the basic information about a general resource.

There is one record per general resource profile.

Note: For some applications, such as SQL/VM, the start and end positions must account for a 4 position length indicator at the front of each record. For applications such as these, 4 would be added to each of the start and end positions indicated.

Table 173. General Resource Basic Data Record				
Field Name	**Type**	**Position**		**Comments**
		Start	**End**	
GRBD_RECORD_TYPE	Int	1	4	Record type of the General Resource Basic Data record (0500).
GRBD_NAME	Char	6	251	General resource name as taken from the profile name.
GRBD_CLASS_NAME	Char	253	260	Name of the class to which the general resource profile belongs.
GRBD_GENERIC	Yes/No	262	265	Is this a generic profile?
GRBD_CLASS	Int	267	269	The class number of the profile.
GRBD_CREATE_DATE	Date	271	280	Date the profile was created.
GRBD_OWNER_ID	Char	282	289	The user ID or group ID which owns the profile.
GRBD_LASTREF_DATE	Date	291	300	The date that the resource was last referenced.
GRBD_LASTCHG_DATE	Date	302	311	The date that the resource was last changed.

Table 173. General Resource Basic Data Record (continued)

Field Name	Type	Position		Comments
		Start	End	
GRBD_ALTER_CNT	Int	313	317	The number of times that the resource was accessed with ALTER authority.
GRBD_CONTROL_CNT	Int	319	323	The number of times that the resource was accessed with CONTROL authority.
GRBD_UPDATE_CNT	Int	325	329	The number of times that the resource was accessed with UPDATE authority.
GRBD_READ_CNT	Int	331	335	The number of times that the resource was accessed with READ authority.
GRBD_UACC	Char	337	344	The universal access of this resource. Valid values are NONE, READ, EXECUTE, UPDATE, CONTROL, and ALTER.
GRBD_AUDIT_LEVEL	Char	346	353	Indicates the level of resource-owner-specified auditing that is performed. Valid values are ALL, SUCCESS, FAIL, and NONE.
GRBD_LEVEL	Int	355	357	The level of the resource.
GRBD_GAUDIT_LEVEL	Char	359	366	Indicates the level of auditor-specified auditing that is performed. Valid values are ALL, SUCCESS, FAIL, and NONE.
GRBD_INSTALL_DATA	Char	368	622	Installation defined data.
GRBD_AUDIT_OKQUAL	Char	624	631	The resource-owner-specified successful access audit qualifier. This is set to blanks if AUDIT_LEVEL is NONE. Otherwise, it is set to either READ, UPDATE, CONTROL, or ALTER.
GRBD_AUDIT_FAQUAL	Char	633	640	The resource-owner-specified failing access audit qualifier. This is set to blanks if AUDIT_LEVEL is NONE. Otherwise, it is set to either READ, UPDATE, CONTROL, or ALTER.
GRBD_GAUDIT_OKQUAL	Char	642	649	The auditor-specified successful access audit qualifier. This is set to blanks if GAUDIT_LEVEL is NONE. Otherwise, it is set to either READ, UPDATE, CONTROL, or ALTER.
GRBD_GAUDIT_FAQUAL	Char	651	658	The auditor-specified failing access audit qualifier. This is set to blanks if GAUDIT_LEVEL is NONE. Otherwise, it is set to either READ, UPDATE, CONTROL, or ALTER.
GRBD_WARNING	Yes/No	660	663	Does this resource have the WARNING attribute?
GRBD_SINGLEDS	Yes/No	665	668	If this is a TAPEVOL profile, is there only one data set on this tape?
GRBD_AUTO	Yes/No	670	673	If this is a TAPEVOL profile, is the TAPEVOL protection automatic?
GRBD_TVTOC	Yes/No	675	678	If this is a TAPEVOL profile, is there a tape volume table of contents on this tape?
GRBD_NOTIFY_ID	Char	680	687	User ID that is notified when violations occur.
GRBD_ACCESS_SUN	Yes/No	689	692	Can the terminal be used on Sunday?
GRBD_ACCESS_MON	Yes/No	694	697	Can the terminal be used on Monday?
GRBD_ACCESS_TUE	Yes/No	699	702	Can the terminal be used on Tuesday?
GRBD_ACCESS_WED	Yes/No	704	707	Can the terminal be used on Wednesday?
GRBD_ACCESS_THU	Yes/No	709	712	Can the terminal be used on Thursday?
GRBD_ACCESS_FRI	Yes/No	714	717	Can the terminal be used on Friday?
GRBD_ACCESS_SAT	Yes/No	719	722	Can the terminal be used on Saturday?
GRBD_START_TIME	Time	724	731	After what time may a user logon from this terminal?
GRBD_END_TIME	Time	733	740	After what time may a user not logon from this terminal?

Table 173. General Resource Basic Data Record (continued)				
Field Name	**Type**	**Position**		**Comments**
		Start	**End**	
GRBD_ZONE_OFFSET	Char	742	746	Time zone in which the terminal is located. Expressed as hh:mm. Blank if the time zone has not been specified.
GRBD_ZONE_DIRECT	Char	748	748	The direction of the time zone shift. Valid values are E(east), W(west), and blank.
GRBD_SECLEVEL	Int	750	752	The security level of the general resource.
GRBD_APPL_DATA	Char	754	1,008	Installation defined data.
GRBD_SECLABEL	Char	1,010	1,017	The security label for the general resource.

General Resource Tape Volume Data Record

The General Resource Tape Volume Data Record defines the characteristics of the tape volume upon which a data set resides.

There is one record per general resource/tape volume combination.

Note: For some applications, such as SQL/VM, the start and end positions must account for a 4 position length indicator at the front of each record. For applications such as these, 4 would be added to each of the start and end positions indicated.

Table 174. General Resource Tape Volume Record				
Field Name	**Type**	**Position**		**Comments**
		Start	**End**	
GRTVOL_RECORD_TYPE	Int	1	4	Record type of the General Resource Tape Volume Data record (0501).
GRTVOL_NAME	Char	6	251	General resource name as taken from the profile name.
GRTVOL_CLASS_NAME	Char	253	260	Name of the class to which the general resource profile belongs, namely TAPEVOL.
GRTVOL_SEQUENCE	Int	262	266	The file sequence number of the tape data set.
GRTVOL_CREATE_DATE	Date	268	277	Creation date of the tape data set.
GRTVOL_DISCRETE	Yes/No	279	282	Does a discrete profile exist?
GRTVOL_INTERN_NAME	Char	284	327	The RACF internal data set name.
GRTVOL_INTERN_VOLS	Char	329	583	The volumes upon which the data set resides.
GRTVOL_CREATE_NAME	Char	585	628	The data set name used when creating the data set.

General Resource Categories Record

The General Resource Categories record defines the categories associated with a general resource.

There is one record per general resource/category combination.

Note: For some applications, such as SQL/VM, the start and end positions must account for a 4 position length indicator at the front of each record. For applications such as these, 4 would be added to each of the start and end positions indicated.

Table 175. General Resource Categories Record				
Field Name	**Type**	**Position**		**Comments**
		Start	**End**	
GRCAT_RECORD_TYPE	Int	1	4	Record type of the General Resources Categories record (0502).
GRCAT_NAME	Char	6	251	General resource name as taken from the profile name.

Table 175. General Resource Categories Record (continued)				
Field Name	**Type**	**Position**		**Comments**
		Start	**End**	
GRCAT_CLASS_NAME	Char	253	260	Name of the class to which the general resource profile belongs.
GRCAT_CATEGORY	Int	262	266	Category to which this general resource belongs.

General Resource Members Record

The General Resource Members record defines the members of a general resource profile group.

There is one record per general resource/member combination.

Note: For some applications, such as SQL/VM, the start and end positions must account for a 4 position length indicator at the front of each record. For applications such as these, 4 would be added to each of the start and end positions indicated.

Table 176. General Resource Members Record				
Field Name	**Type**	**Position**		**Comments**
		Start	**End**	
GRMEM_RECORD_TYPE	Int	1	4	Record type of the General Resource Members record (0503).
GRMEM_NAME	Char	6	251	General resource name as taken from the profile name.
GRMEM_CLASS_NAME	Char	253	260	Name of the class to which the general resource profile belongs.
GRMEM_MEMBER	Char	262	516	Member value for this general resource. • For VMXEVENT profiles, this is the element that is being audited. • For PROGRAM profiles, this is the name of the data set which contains the program. • For GLOBAL profiles, this is the name of the resource for which a global access applies. • For SECDATA security level (SECLEVEL) profiles, this is the level name. For SECDATA CATEGORY profiles, this is the category name. • For NODES profiles, this is the user ID, group ID, and SECLABEL translation data.
GRMEM_GLOBAL_ACC	Char	518	525	If this is a GLOBAL profile, this is the access that is allowed. Valid values are NONE, READ, UPDATE, CONTROL, and ALTER.
GRMEM_PADS_DATA	Char	527	534	If this is a PROGRAM profile, this field contains the Program Access to Data Set (PADS) information for the profile. Valid values are PADCHK and NOPADCHK.
GRMEM_VOL_NAME	Char	536	541	If this is a PROGRAM profile, this field defines the volume upon which the program resides.
GRMEM_VMXEVENT_DATA	Char	543	547	If this is a VMXEVENT profile, this field defines the level of auditing that is being performed. Valid values are CTL, AUDIT, and NOCTL.
GRMEM_SECLEVEL	Int	549	553	If this is a SECLEVEL profile in the SECDATA class, this is the numeric security level that is associated with the SECLEVEL.
GRMEM_CATEGORY	Int	555	559	If this is a CATEGORY profile in the SECDATA class, this is the numeric category that is associated with the CATEGORY.

General Resource Volumes Record

The General Resource Volumes record defines the volumes in a tape volume set.

There is one record per tape volume set/volume combination.

Note: For some applications, such as SQL/VM, the start and end positions must account for a 4 position length indicator at the front of each record. For applications such as these, 4 would be added to each of the start and end positions indicated.

Table 177. General Resource Volumes Record

Field Name	Type	Position		Comments
		Start	End	
GRVOL_RECORD_TYPE	Int	1	4	Record type of the General Resources Volumes record (0504).
GRVOL_NAME	Char	6	251	General resource name as taken from the profile name.
GRVOL_CLASS_NAME	Char	253	260	Name of the class to which the general resource profile belongs, namely TAPEVOL.
GRVOL_VOL_NAME	Char	262	267	Name of a volume in a tape volume set.

General Resource Access Record

The General Resource Access record defines the users or groups who have specific access to general resources.

There is one record per general resource/authorization combination.

Note: For some applications, such as SQL/VM, the start and end positions must account for a 4 position length indicator at the front of each record. For applications such as these, 4 would be added to each of the start and end positions indicated.

Table 178. General Resource Access Record

Field Name	Type	Position		Comments
		Start	End	
GRACC_RECORD_TYPE	Int	1	4	Record type of the General Resource Access record (0505).
GRACC_NAME	Char	6	251	General resource name as taken from the profile name.
GRACC_CLASS_NAME	Char	253	260	Name of the class to which the general resource profile belongs.
GRACC_AUTH_ID	Char	262	269	User ID or group ID which is authorized to use the general resource.
GRACC_ACCESS	Char	271	278	The authority that the user or group has over the resource. Valid values are NONE, READ, UPDATE, CONTROL, and ALTER.
GRACC_ACCESS_CNT	Int	280	284	The number of times that the resource was accessed.

General Resource Installation Data Record

The General Resource Installation Data record defines the user data associated with a general resource.

There is one record per general resource/data combination.

This record type contains data stored in the USRCNT repeat group, which is a field in the RACF database that is reserved for your installation's use. None of the RACF commands manipulate this field. Do not confuse this field with the GRBD_INSTALL_DATA field, shown in Table 173 on page 293, which you enter into the database using the RDEFINE and RALTER commands.

Note: For some applications, such as SQL/VM, the start and end positions must account for a 4 position length indicator at the front of each record. For applications such as these, 4 would be added to each of the start and end positions indicated.

Table 179. General Resource Installation Data Record

Field Name	Type	Position		Comments
		Start	End	
GRINSTD_RECORD_TYPE	Int	1	4	Record type of the General Resource Installation Data record (0506).
GRINSTD_NAME	Char	6	251	General resource name as taken from the profile name.
GRINSTD_CLASS_NAME	Char	253	260	Name of the class to which the general resource profile belongs.
GRINSTD_USR_NAME	Char	262	269	The name of the installation defined field.
GRINSTD_USR_DATA	Char	271	525	The data for the installation defined field.
GRINSTD_USR_FLAG	Char	527	534	The flag for the installation defined field in the form X<nn>.

General Resource Conditional Access Record

The General Resource Conditional Access record defines the conditional access to a general resource.

There is one record per general resource/access combination.

Note: For some applications, such as SQL/VM, the start and end positions must account for a 4 position length indicator at the front of each record. For applications such as these, 4 would be added to each of the start and end positions indicated.

Table 180. General Resource Conditional Access Record

Field Name	Type	Position		Comments
		Start	End	
GRCACC_RECORD_TYPE	Int	1	4	Record type of the General Resources Conditional Access record (0507).
GRCACC_NAME	Char	6	251	General resource name as taken from the profile name.
GRCACC_CLASS_NAME	Char	253	260	Name of the class to which the general resource profile belongs.
GRCACC_CATYPE	Char	262	269	The type of conditional access checking that is being performed. Valid values are `CONSOLE`, `TERMINAL` and `JESINPUT`.
GRCACC_CANAME	Char	271	278	The name of a conditional access element which is permitted access.
GRCACC_AUTH_ID	Char	280	287	The user ID or group ID which has authority to the general resource.
GRCACC_ACCESS	Char	289	296	The authority of the conditional access element/user combination. Valid values are `NONE`, `READ`, `UPDATE`, `CONTROL`, and `ALTER`.
GRCACC_ACCESS_CNT	Int	298	302	The number of times that the general resource was accessed.

General Resource Session Data Record

The General Resource Session Data record defines the session data associated with a general resource.

There is one record per APPCLU profile.

Note: For some applications, such as SQL/VM, the start and end positions must account for a 4 position length indicator at the front of each record. For applications such as these, 4 would be added to each of the start and end positions indicated.

Table 181. General Resource Session Data Record				
Field Name	**Type**	**Position**		**Comments**
		Start	**End**	
GRSES_RECORD_TYPE	Int	1	4	Record type of the General Resources Session Data record (0510).
GRSES_NAME	Char	6	251	General resource name as taken from the profile name.
GRSES_CLASS_NAME	Char	253	260	Name of the class to which the general resource profile belongs, namely APPCLU.
GRSES_SESSION_KEY	Char	262	269	The key associated with the APPC session.
GRSES_LOCKED	Yes/No	271	274	Is the profile locked?
GRSES_KEY_DATE	Date	276	285	Last date that the session key was changed.
GRSES_KEY_INTERVAL	Int	287	291	Number of days that the key is valid.
GRSES_SLS_FAIL	Int	293	297	Current number of failed attempts.
GRSES_MAX_FAIL	Int	299	303	Number of failed attempts before lockout.
GRSES_CONVSEC	Char	305	312	Specifies the security checking performed when sessions are established. Valid values are NONE, CONVSEC, PERSISTV, ALREADYV, and AVPV.

General Resource Session Entities

The General Resource Session Entities record defines the entities associated with a general resource APPCLU profile.

There is one record per APPCLU profile/session entity combination.

Note: For some applications, such as SQL/VM, the start and end positions must account for a 4 position length indicator at the front of each record. For applications such as these, 4 would be added to each of the start and end positions indicated.

Table 182. General Resource Session Entity Record				
Field Name	**Type**	**Position**		**Comments**
		Start	**End**	
GRSESE_RECORD_TYPE	Int	1	4	Record type of the General Resources Session Entities record (0511).
GRSESE_NAME	Char	6	251	General resource name as taken from the profile name.
GRSESE_CLASS_NAME	Char	253	260	Name of the class to which the general resource profile belongs, namely APPCLU.
GRSESE_ENTITY_NAME	Char	262	296	Entity name.
GRSES_FAIL_CNT	Int	298	302	The number of failed session attempts.

General Resource DLF Data Record

The General Resource DLF Data record defines the Data Lookaside Facility (DLF) data associated with a general resource.

There is one record per general resource/DLF data combination.

Note: For some applications, such as SQL/VM, the start and end positions must account for a 4 position length indicator at the front of each record. For applications such as these, 4 would be added to each of the start and end positions indicated.

Table 183. General Resource DLF Data Record				
Field Name	**Type**	**Position**		**Comments**
		Start	**End**	
GRDLF_RECORD_TYPE	Int	1	4	Record type of the General Resources DLF Data record (0520).
GRDLF_NAME	Char	6	251	General resource name as taken from the profile name.
GRDLF_CLASS_NAME	Char	253	260	Name of the class to which the general resource profile belongs, namely DLFCLASS.
GRDLF_RETAIN	Yes/No	262	265	Is this a retained resource?

General Resource DLF Job Names Record

The General Resource DLF Job Names record defines the job names associated with a DLF general resource.

There is one record per general resource/DLF job name combination.

Note: For some applications, such as SQL/VM, the start and end positions must account for a 4 position length indicator at the front of each record. For applications such as these, 4 would be added to each of the start and end positions indicated.

Table 184. General Resource DLF Job Names Record				
Field Name	**Type**	**Position**		**Comments**
		Start	**End**	
GRDLFJ_RECORD_TYPE	Int	1	4	Record type of the General Resources DLF Job Names record (0521).
GRDLFJ_NAME	Char	6	251	General resource name as taken from the profile name.
GRDLFJ_CLASS_NAME	Char	253	260	Name of the class to which the general resource profile belongs, namely DLFCLASS.
GRDLFJ_JOB_NAME	Char	262	269	The job name associated with the general resource.

Chapter 6. The RACF Secured Signon PassTicket

The RACF secured signon function provides an alternative to the RACF password called a *PassTicket*. The RACF PassTicket is a *one-time-only* password that is generated by a requesting product or function. It is an alternative to the RACF password that removes the need to send RACF passwords across the network in clear text. It makes it possible to move the authentication of a mainframe application user ID from RACF to another authorized function executing on the host system or to the work station local area network (LAN) environment.

Generating a PassTicket

A product or function that generates a PassTicket must use the RACF PassTicket generator algorithm. This algorithm requires specific information as input data and produces a PassTicket that substitutes for a specific end-user RACF password. RACF uses the PassTicket to authenticate the end-user for a specific application running on a specific system that uses RACF for identification and authentication.

There are two ways to generate and evaluate a PassTicket using the algorithm:

- For any function that generates a PassTicket, you can create a program that incorporates the algorithm. See “Incorporating the PassTicket Generator Algorithm into Your Program” on page 301.
- You can use DIAGNOSE X'A0' subcode X'54'. For more information, see Chapter 2, “Diagnose X'A0' Subcodes,” on page 19

Incorporating the PassTicket Generator Algorithm into Your Program

To generate a PassTicket without using the RACF callable service, you need to incorporate the RACF PassTicket generator algorithm into your program.

The RACF PassTicket algorithm consists of two parts:

- The RACF PassTicket generator
- The RACF PassTicket time-coder

The time-coder is invoked from within the RACF PassTicket generator and returns its results to the generator.

The flowcharts in Figure 5 on page 302 and Figure 6 on page 303 and the descriptions that follow show how to implement the RACF PassTicket generator algorithm.

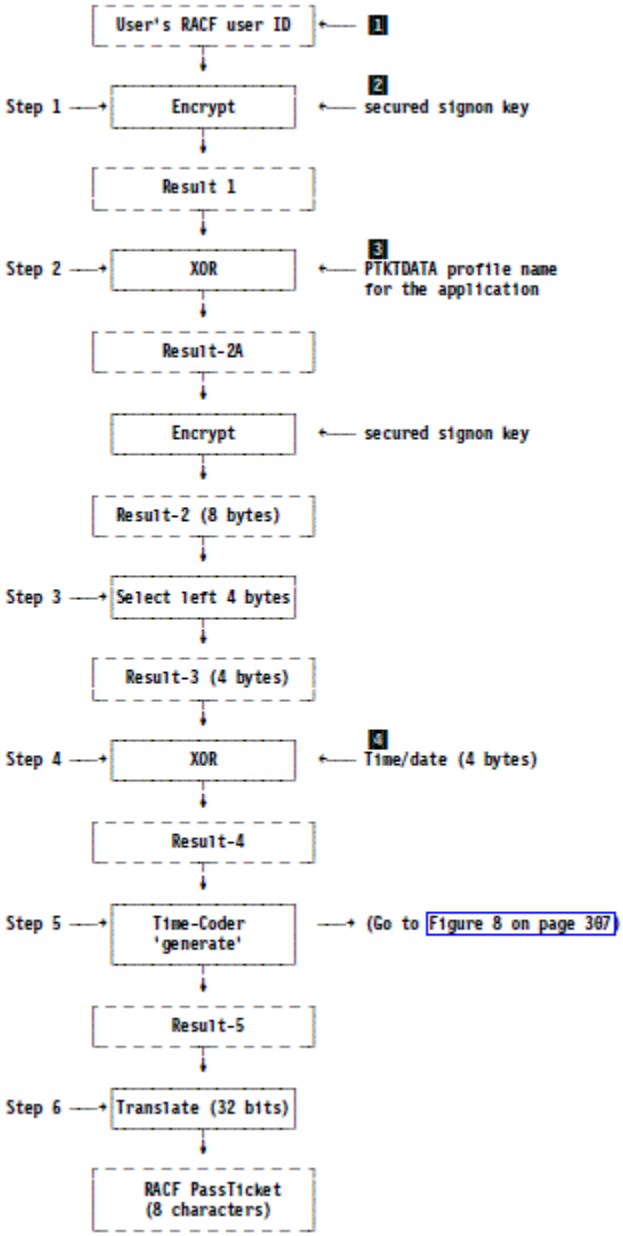


Figure 5. RACF PassTicket Generator for secured signon

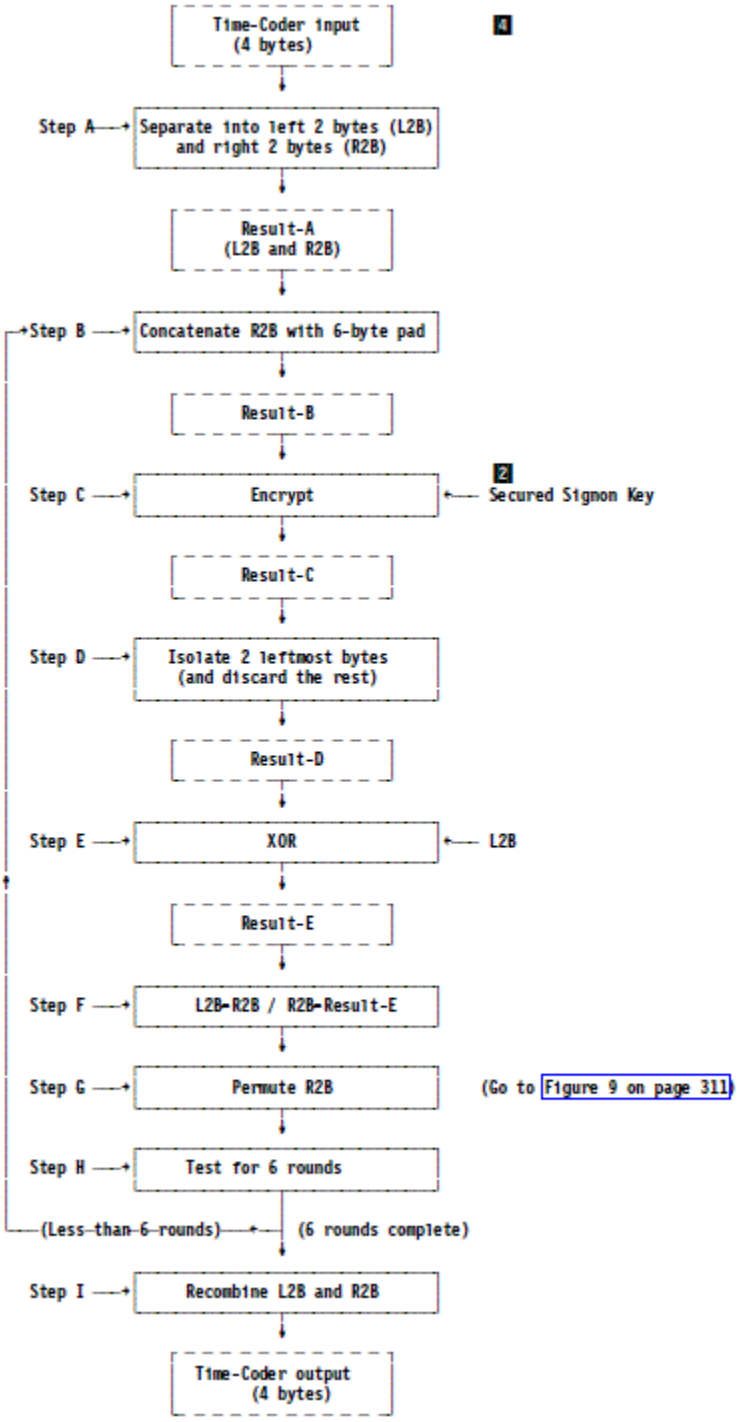


Figure 6. Algorithm for RACF PassTicket Time-Coder used for the secured signon

Input Data for the Algorithm

To successfully use the PassTicket, the target application using RACF to identify and authenticate a user ID needs to have specific information for processing according to the algorithm. As shown in Figure 6 on page 303, these are:

- A RACF host user ID
- The RACF secured signon application key
- The PTKTDATA class profile name

- Time and date information

1

The RACF user ID Is an ID that:

- Identifies the user ID on the system on which the target application runs
- Is represented in EBCDIC
- Is left-justified and padded with blanks on the right to a length of 8 bytes

2

The RACF secured signon application key Is an 8-byte (64-bit) string that:

- Must match the key value used when defining the application to the PTKTDATA class to RACF
- Contains only the characters 0 through 9 and A through F

3

The PTKTDATA class profile name Is the name of a RACF profile in the PTKTDATA class as defined for a particular application. You can use it to associate a secured signon key with a particular host application. See *z/VM: RACF Security Server Security Administrator's Guide* for information on profiles.

The name:

- Is represented in EBCDIC
- Is left-justified and padded with blanks on the right to a length of 8 bytes

4

Time and date information Is the required time and date input data from the application that is providing the logon function. This information:

- Must be an 4-byte binary number
- Shows how many seconds have elapsed since January 1, 1970, at 0000 Greenwich Mean Time (GMT)

Several programming languages support a function for representing time in this way. In C language, for example, you can obtain the time in this way:

1. Declare the variable *ts* as **`long`**.
2. Invoke the function **`time(&ts)`**.

This produces the number of seconds that have elapsed since January 1, 1970 at 0000 GMT, expressed as an unsigned long integer.

Note:

1. It is likely that the computer that authenticates the PassTicket is not the computer that generated it. To provide for differences in their internal clocks, the algorithm allows the generated time to be 10 minutes on either side of the TOD clock of the computer that is evaluating the PassTicket.
2. For RACF to properly evaluate PassTickets, the TOD clock must be properly set to GMT rather than local time.

How the Generator Algorithm Works

The RACF PassTicket generator algorithm uses the input information to create a PassTicket. By using cryptographic techniques, the algorithm ensures that each PassTicket is unpredictable.

The PassTicket is an 8-character alphanumeric string that can contain the characters A through Z and 0 through 9. The actual PassTicket depends on the input values.

The following steps describe this process (see Figure 5 on page 302 for a summary):

1. The RACF user ID 1 is encrypted using the RACF secured signon application key 2 as the encryption key to produce Result-1.

   Note: All encryptions use the U.S. National Institute of Standards and Technology Data Encryption Standard (DES) algorithm. Only the DES algorithm encoding is involved. You cannot perform general encryption and decryption of data with this implementation.

2. Result-1 from the first encryption is XORed[1] with the PTKTDATA profile name 3. The PTKTDATA profile name must be 8 bytes of EBCDIC characters with trailing blanks. The result (Result-2A) is encrypted using the application key value 2 as the encryption key to produce Result-2.

   Note: If you understand cryptographic techniques, you should recognize the flow (Steps 1 and 2) to be a common cryptographic architecture (CCA) standard message authentication code algorithm.

3. The left 4 bytes from Result-2 of the second encryption are selected as input to the next step. The rest are discarded.
4. The resulting 4 bytes (Result-3) are XORed with the time and date information 4. The time and date is in the form of a 4-byte field that contains the number of seconds that have elapsed since January 1, 1970 at 0000 GMT in the form of a binary integer. (See “Input Data for the Algorithm” on page 303 for a complete description.)
5. The result (Result-4) of that procedure is passed to the time-coder routine. Refer to the diagram in Figure 6 on page 303 and to “How the Time-Coder Algorithm Works” on page 305 to understand that process.
6. The result (Result-5) of the time-coder routine is translated, using a translation table described in “The Translation Table” on page 307, to an 8-character string called the PassTicket. It is used in the user's host application signon request instead of the user's regular RACF password.

How the Time-Coder Algorithm Works

The RACF PassTicket time-coder algorithm uses the result of Step “4” on page 305 of the generator algorithm. It creates the time-coder information and passes it back to step “6” on page 305 of that algorithm.

The following steps, which make up Step “5” on page 305 of the generator algorithm, shown in Figure 6 on page 303 describe this process:

Step A
Separate the 4-byte time-coder input (Result-4) into two portions, L2B (the left side), and R2B (the right side) to produce Result-A.

Step B

- Concatenate R2B (the right 2 bytes from Result-A) with 6 bytes of padding bits to form Result-B. In the resulting 8-byte string, the 2 bytes of R2B occupy the leftmost 2 byte positions.

  The padding bits consist of two separate 6 byte strings: PAD1 and PAD2. PAD1 is the left half and PAD2 is the right half of a 12 byte string consisting of the user ID (from Step “1” on page 305 in “How the Generator Algorithm Works” on page 304) left justified and padded to the right with hexadecimal '55's. For example, if the user ID is "TOM", PAD1 is 'E3D6D4555555' and PAD2 is '555555555555'. If the user ID is "IBMUSER", PAD1 is 'C9C2D4E4D2C5' and PAD2 is 'D95555555555'. PAD1 is used for time coder loop rounds 1, 3, and 5. PAD2 is used for time coder loop rounds 2, 4, and 6.

Step C
Result-B is encrypted using the RACF secured signon application key 2 as the encryption key to produce Result-C.

[1] XOR is a Boolean function that processes two-bit strings of the same length, producing a third string of the same length as the output. In the output string, a bit is ON if the corresponding bit is ON for one of the input bit strings, but not both.

Step D
The left 2 bytes from the Result-C are isolated and the rest of the value is discarded, producing Result-D.

Step E
Result-D is XORed with L2B (from Result-A) to produce Result-E.

Step F
The values of L2B and R2B are redefined:

1. L2B is set equal to R2B.
2. R2B is set equal to Result-E.

Step G
R2B is permuted[2] using the permutation tables in Figure 7 on page 307, where the table used reflects the number of the round. For example, for the first time through, R2B is permuted using table 1.

Step H
This step counts the number of time-coder rounds that have been completed. If the value is less than 6, the time-coder returns to Step b for another round. If 6 rounds have been completed, processing continues with the next step.

Step I
L2B (left 2 bytes) and R2B (right 2 bytes) are recombined into a 32-bit string. This completes the time-coder processing and produces Result-5. This result is passed back to the generator algorithm as input to Step “6” on page 305 for translation.

The Permutation Tables

A permutation table exists for each round of permutations that occurs during the time-coder process.

The six permutation tables work in the following manner:

- The upper of the two rows of numbers (O=>) represents the output positions, from left to right, of the 16 bits being permuted.
- The lower of the two rows (I=>) represents the input-bit position.

For example, using Permutation Table 1:

- Output-bit position 1 consists of the bit (on or off) in input-bit position 10.
- Output-bit position 2 consists of the bit in input-bit position 2.

[2] To permute is to transform or change the order of members of a group.

```
                    Permutation Table 1

O=>  1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16
I=> 10   2  12   4  14   6  16   8   9   1  11   3  13   5  15   7



                    Permutation Table 2

O=>  1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16
I=>  1  10   3  12  13  16   7  15   9   2  11   4   5  14   8   6



                    Permutation Table 3

O=>  1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16
I=>  3  10   1  12  13  16   9  15   7   2  14   4   5  11   8   6



                    Permutation Table 4

O=>  1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16
I=> 10   4  12   2  14   8  16   6   9   1  13   3  11   5  15   7



                    Permutation Table 5

O=>  1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16
I=>  4  10  12   1   8  16  14   5   9   2  13   3  11   7  15   6



                    Permutation Table 6

O=>  1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16
I=>  1  16  15  14  13  12  11  10   9   8   7   6   5   4   3   2
```

Figure 7. Permutation Tables for RACF secured signon

The Translation Table

The translation table consists of 36 slots. The first 26 slots are occupied by the letters of the alphabet: A–Z. The last ten slots are occupied by the numerics 0–9.

```
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
A B C D E F G H I J  K  L  M  N  O  P  Q  R  S  T  U  V  W  X  Y  Z
26 27 28 29 30 31 32 33 34 35
0  1  2  3  4  5  6  7  8  9
```

Figure 8. Translation Table for RACF secured signon

The Translation Process

The time-coder output produced by the process described in Figure 6 on page 303 is translated into 8 alphanumeric characters in the following manner:

1. Bits 31, 32, 1, 2, 3, and 4 are translated to PassTicket character position 1, which is the leftmost position in the 8-byte alphanumeric PassTicket field.

   To produce this character:

   - The binary number, represented by the six bits, is divided by decimal 36.
   - The remainder is used as an index into the translation table.

   For example, a remainder of 0 translates to a PassTicket character of A and a remainder of 20 translates to a PassTicket character of U.
2. The process is repeated with the rest of the bit string:
   - Bits 3 through 8 are translated to PassTicket character position 2.

- Bits 7 through 12 are translated to PassTicket character position 3.
- Bits 11 through 16 are translated to PassTicket character position 4.
- Bits 15 through 20 are translated to PassTicket character position 5.
- Bits 19 through 24 are translated to PassTicket character position 6.
- Bits 23 through 28 are translated to PassTicket character position 7.
- Bits 27 through 32 are translated to PassTicket character position 8.

Example

In this example, the result of the time-coder step is X'07247F79'. Using that value, a PassTicket is generated as follows:

```
Byte                1        2        3        4
-----------------------------------------------------
Hexadecimal value   07       24       7F       79
-----------------------------------------------------
Binary            00000111 00100100 01111111 01111001
-----------------------------------------------------
Bit Position      00000000 01111111 11122222 22222333
                  12345678 90123456 78901234 56789012
-----------------------------------------------------
 PassTicket                                    Translates
  Character                                     to
  Position   Binary    Integer       Remainder   Character
-----------------------------------------------------
        1    010000      16  x 1/36 => 16         Q
        2    000111       7  x 1/36 =>  7         H
        3    110010      50  x 1/36 => 14         O
        4    100100      36  x 1/36 =>  0         A
        5    000111       7  x 1/36 =>  7         H
        6    111111      63  x 1/36 => 27         1
        7    110111      55  x 1/36 => 19         T
        8    111001      57  x 1/36 => 21         V
-----------------------------------------------------
```

1. Bits 31, 32, 1, 2, 3, and 4 (6 bits total) are translated to produce the PassTicket character in position 1.

   The six bits (binary '010000' or decimal 16) are divided by decimal 36.
2. The remainder (decimal 16) becomes the index into the translation table. The result is character 'Q'.
3. Repeat the process for the rest of the bits.
   - Bits 3 through 8 are translated to PassTicket character 'H'.
   - Bits 7 through 12 are translated to PassTicket character 'O'.
   - Bits 11 through 16 are translated to PassTicket character 'A'.
   - Bits 15 through 20 are translated to PassTicket character 'H'.
   - Bits 19 through 24 are translated to PassTicket character '1'.
   - Bits 23 through 28 are translated to PassTicket character 'T'.
   - Bits 27 through 32 are translated to PassTicket character 'V'.

The resulting PassTicket returned as output is QHOAH1TV.

Appendix A. Date Conversion Routine

RACF provides installations with the IRRDCR00 module, which is the date conversion routine that enables programs to specify and identify dates beyond the end of the 20th century.

Note: This routine can only be invoked from within the RACF service machine, that is, from within a RACF exit.

Using this routine, RACF exit programs can convert a three-byte packed-decimal date to a four-byte packed-decimal date. The three-byte date has the form *yydddF*, and the four-byte date has the form *ccyydddF*, where *cc* is 00 for years 1971 through 1999 and is 01 for years 2000 through 2070. In the four-byte form, the routine interprets the year as 19*yy* when *yy* is 71 or higher and as 20*yy* when *yy* is less than 71.

This routine resides in the LPA. The system loads the address of this routine in RCVTDATP and sets bit RCVTD4OK (X'08') in flag byte RCVTMFLG during RACF initialization. The routine's caller can use the address in RCVTDATP.

Invoking the Date Conversion Routine

When a program invokes the date conversion routine, the program must pass two parameters to it:

- **Three-byte date**—a three-byte field containing a packed-decimal date (format *yydddF*).
- **Four-byte date**—a four-byte field

Note: This routine runs in the caller's mode, state, and key. Recovery is handled by the calling program.

Format of Returned Converted Date

The routine returns a four-byte packed-decimal date whose format is either *00yydddF* (for 1971-1999) or *01yydddF* (for 2000-2070).

If *ddd* is 000 in the three-byte *yydddF* date field passed to the routine, the routine returns 00 for *cc* (indicating a year 19*yy*), regardless of what *yy* is.

Return Code

The return code from this routine follows RACF conventions with a return code of 00 (X'00') indicating successful date conversion. This code is returned in register 15.

The return code is as follows:

Note to Reader
The return code is shown in hexadecimal.

Return Code
Meaning

00
The date conversion function completed successfully.

Appendix B. ICHEINTY, ICHETEST, and ICHEACTN Macros

Note: You can only use the ICHEINTY, ICHETEST, and ICHEACTN macros in the RACF service machine, for example, from an installation exit. You cannot use the ICHEINTY, ICHETEST, and ICHEACTN macros from a user's machine.

The ICHEINTY, ICHETEST, ICHEACTN product macros are described in this appendix, rather than in the body of the book, because of their complexity and the cautions required in their use. IBM recommends the use of the RACROUTE REQUEST=EXTRACT macro instead for the cases where it can be used. However, only the RACF command processors do complete validation of the data entering the database, so it is better to use RACF commands than either ICHEINTY or RACROUTE REQUEST=EXTRACT, see *z/VM: Security Server RACROUTE Macro Reference*.

In general, you should use the RACF command processors to create RACF resource profiles. If you use ICHEINTY instead, you should create profiles which are supported by the command processors. For instance, ICHEINTY allows you to create a fully-qualified generic profile in a general resource class and a data set profile containing characters that are not valid, but those profiles are not supported by the RACF command processors.

You can use the ICHEINTY, ICHETEST, and ICHEACTN macros to locate (retrieve) and update the various profiles on the RACF database.

ICHEINTY
Locates or updates the profile.

ICHETEST
Tests for user-specified conditions on selected fields in the profile.

ICHEACTN
Retrieves or alters specified fields within the retrieved profile.

Installations planning to use these macros should exercise caution because the ICHEINTY, ICHETEST, and ICHEACTN macros:

- Perform only limited parameter validation. The module issuing these macros must be authorized (supervisor state, system key, or APF-authorized).
- Do not pass control to any exit routines except indirectly: if FLDACC=YES has been specified on the ICHEINTY macro, then the RACHECK request exits will be given control during field access checking.
- Do not do any logging except indirectly: logging can occur during field access checking if the RACHECK request exit requests it.
- Do not complete data consistency checking. For example,
  1. They do not ensure that all fields in a profile will have the data expected by subsequent RACF processing.
  2. They do not ensure that related profiles are updated in a consistent manner. For example, a group profile must point to its superior group profile and the superior group must point to the subgroup profile.

Beginning with RACF 1.8, these three macros have several functional and usability enhancements. Existing code that uses ICHEINTY, ICHETEST, and ICHEACTN will continue to work as before. It is only when you specify RELEASE=1.8 or later that the processing for each of these macros changes.

Note: Readers of this section should be very familiar with the template information contained in Appendix F, “RACF Database Templates,” on page 377.

These macros may be used by callers in either 31- or 24-bit addressing mode. The parameter lists may be located above 16MB if the caller is in 31-bit mode.

ICHEINTY Macro

The ICHEINTY macro provides a direct interface to the RACF database through the RACF manager. Its function is to locate and/or update a profile in the RACF database.

You can use the ICHEINTY macro with the ICHETEST and ICHEACTN macros to test and conditionally update fields in RACF profiles.

The ICHEINTY macro must be issued from a task running in non-cross-memory mode with no locks held. The issuing task must be authorized (APF-authorized, in system key 0-7, or running in supervisor state).

You may reference only one segment with each ICHEINTY call; however, you may access more than one field in a segment using a single call. If you need to retrieve or update more than one segment, issue a separate ICHEINTY for each segment.

The format of the ICHEINTY macro definition is:

```
[label] ICHEINTY [operation]
                 [,TYPE='GRP'|'USR'|'CON'|'DS'|'GEN']
                 [,ENTRY=entry-address]
                 [,ENTRYX=extended-entry address]
                 [,CLASS=class-address]
                 [,FLDACC=NO|YES]
                 [,RELEASE=number|(,CHECK)|(number,CHECK)]
                 [,RUN=YES|NO]
                 [,ACEE=acee address]
                 [,WKSP=subpool number]
                 [,CHAIN=parm-list address]
                 [,DATAMAP=OLD|NEW]
                 [,SEGMENT='segment name']
                 [,VOLUME=volume-address]
                 [,ACTIONS=(action-address,...)]
                 [,TESTS=(test-addr[,[AND],test-addr]...)]
                 [,WKAREA=workarea-address]
                 [,NEWNAME=newname-address]
                 [,NEWNAMX=extended-newname-address]
                 [,RBA=rba-address]
                 [,FLDEF=fldef-address]
                 [,OPTIONS=([NOPRO|TESTM|TESTC|ACTION|
                           NOEXEC|FLDEF],...)]
                 [,SMC=YES|NO]
                 [,GENERIC=NO|YES|UNCOND]
                 [,DATEFMT=YYDDDF|YYYYDDDF]
                 [,MF=I|L|(E,address)]
```

operation

specifies the operation that RACF is to perform on the specified profile. The valid operation values are ADD, ALTER, ALTERI, DELETE, DELETEA, FLDEF, LOCATE, NEXT, NEXTC, and RENAME. This operand is positional and is required if you specify MF=I or MF=L.

Some operations are based on assumptions. If a requested operation violates an assumption, the operation fails.

ADD

defines entities to RACF by adding profiles to the RACF database. ADD processing:

- Creates a profile for the new entry with fields containing null values. (See “Using ICHEACTN to Alter Data When the ICHEINTY Has DATAMAP=NEW” on page 332 and “Using ICHEACTN to Alter Data When ICHEINTY Has DATAMAP=OLD” on page 334.)
- Alters field values as specified by associated ICHEACTN macro instructions.
- Allocates space for the profile on the RACF database and writes the profile.
- Creates an index entry for the profile. The index entry points to the new profile.

Some considerations regarding ADD are:

- ADD processing assumes that the profile does not already exist. If the profile already exists (the index contains the profile name), any of the following conditions causes a return code of 8 (X'08'):

  - TYPE is not ‘DS’
  - TYPE is ‘DS’ and duplicate data set name creation has been prohibited via ICHSECOP.
  - TYPE is ‘DS’ and one of the existing profiles for the entity name contains in its volume list the volume specified by the VOLUME keyword.
- For TYPE=‘DS’, ICHEINTY sets a return code of 60 (X'3C') if VOLUME is not specified and there are multiple profiles for the entity name.
- For TYPE=‘GEN‘, you can add the same entity name to any number of different classes. If the class is TAPEVOL, ICHEINTY sets a return code of 60 (X'3C') if a VOLUME is specified but is not an existing TAPEVOL. ICHEINTY creates a profile for a TAPEVOL only when VOLUME is not specified; otherwise, it updates an existing profile. The macro creates an index entry in either case. The result of this special TAPEVOL processing is that RACF maintains only one profile for a multi-volume tape. You can refer to that profile by specifying any of the volumes it protects.
- You must supply all the information required by RACF for subsequent processing: for example, owner, creation date.
- In general, you should use the RACF command processors to create RACF resource profiles. If you use ICHEINTY instead, you should create profiles which are supported by the command processors. For instance, ICHEINTY allows you to create a fully-qualified generic profile in a general resource class and a data set profile containing characters that are not supported by the RACF command processors.

ALTER
alters field values in an existing profile on the RACF database. ALTER processing:

- Locates the profile on the RACF database.
- Performs the tests that the ICHETEST macros specify, if any macros are present. The TESTS operand on the ICHEINTY macro names the tests to be performed.
- Alters field values as specified by associated ICHEACTN macro instructions.
- Writes the profile back to the primary and backup (if active) RACF databases.

Some considerations regarding ALTER are:

- If the profile is too large to be rewritten to the same location in the RACF database, RACF allocates new space, writes the profile to the new location, and updates the index entry for the profile to point to the new location.

ALTERI
is similar to the ALTER operation with the following exceptions:

- Fields are updated in place. ICHEINTY sets a return code of 68 (X'44') and fails the operation if the altered profile has a length which differs from that of the original profile.
- The update to the field occurs with only a shared lock on the RACF database. Therefore, other ALTERI, LOCATE, NEXT, or NEXTC requests can take place simultaneously.
- You can specify the RBA of the level one index block, containing the pointer to the profile to be altered. This improves processing efficiency.
- ALTERI processing writes the profile back to the primary RACF database only; it does not write to the backup, unless you have otherwise specified in the database name table (ICHRDSNT). RACF uses ALTERI to update statistical information in profiles.

DATEFMT=YYYYDDDF|YYDDDF
Specifies the format of the date that you want to extract or replace. If you specify DATEFMT=YYYYDDDF with a LOCATE, NEXT, or NEXTC operation, RACF retrieves date fields in the format ccyydddF where cc=19 or cc=20. If an ADD, ALTER or ALTERI operation is specified, RACF accepts dates in the format ccyydddF where cc=19 or cc=20. When accepting a date as input to place into the database, RACF validates that cc=19 or 20 and that:

- for cc=19, 70 < yy <= 99 and
- for cc=20, 00 <=yy <= 70

If you specify DATEFMT=YYDDDF, RACF retrieves and accepts dates in the normal three byte format.

To specify the DATEFMT keyword, you must specify Release=1.10.

DATEFMT=YYDDDF is the default.

DELETE
deletes a profile from the RACF database. DELETE processing:

- Deletes the index entry for the entity.
- Frees space used for the profile on the RACF database.

Some considerations regarding DELETE are:

- You cannot specify the ACTIONS operand with the DELETE operation.
- For TYPE='DS', ICHEINTY sets a return code of 60 (X'3C') if you specified VOLUME and a profile containing the specified volume in its VOLSER list was not found. ICHEINTY sets a return code of 56 (X'38') if you do not specify VOLUME and there are multiple profiles for the entity name.
- ICHEINTY deletes the profile for a TAPEVOL only when the volume being deleted is the last in its set. Otherwise, the macro deletes the index entry and removes the volume from the VOLSER list.

DELETEA
Deletes all the members of a TAPEVOL set from the RACF database. It is similar to the DELETE operation with the following exception:

- If you specify 'TYPE=GEN' and the class is TAPEVOL, ICHEINTY deletes the profile along with the index entry for each volume in the set.

FLDEF
Builds the area that the FLDEF operand uses. The area contains control information and the list generated by the TESTS and ACTIONS operands.

Some assumptions and considerations regarding FLDEF are:

- FLDEF creates a separate area for ICHEACTN and ICHETEST pointers, which can be referenced from one or more ICHEINTY macros.
- You can maintain the field definition area with the MF=E form of ICHEINTY FLDEF.
- When referencing the field definition area from a remote ICHEINTY, specify FLDEF=field-definition-area, and do **not** specify any of the ACTION, FLDEF, TESTC, and TESTM options on the OPTIONS keyword.

LOCATE
Retrieves zero or more fields from an existing RACF profile in the RACF data set. LOCATE processing:

- Locates the profile in the RACF database.
- Performs the tests that the ICHETEST macros specify, if any macros are present. The TESTS operand on the ICHEINTY macro names the tests to be performed.
- Retrieves field values as specified by associated ICHEACTN macro instructions into the caller-specified work area.

Some assumptions and considerations regarding LOCATE are:

- ICHEINTY sets a return code of 44 (X'2C') if the values being returned are too large for the work area provided and you did not specify the WKSP operand, which would have provided you with an additional work area.
- For TYPE='DS', ICHEINTY sets a return code of 60 (X'3C') if you specify VOLUME and one or more profiles were found for the data set name but none contained the specified volume name in its VOLSER list. ICHEINTY sets a return code of 56 (X'38') if you do not specify VOLUME and there are multiple profiles for the entity name.

- ICHEINTY sets a return code of 52 (X'34') if an ICHETEST macro specified by the TESTS operand failed. The LOCATE operation terminates at this point.

NEXT
Retrieves zero or more fields from the profile whose name follows the name specified by the ENTRY or ENTRYX operand. The NEXT operation updates the area pointed to by the ENTRY or ENTRYX operand with the name of the profile just completed. NEXT processing:

- Locates the profile of the first entity of the specified type that follows the specified entity located in the RACF database.
- Performs the tests that the ICHETEST macros specify, if any macros are present. The TESTS operand on the ICHEINTY macro names the tests to be performed.
- Retrieves field values as specified by associated ICHEACTN macro instructions into the caller specified work area.

Some considerations regarding NEXT are:

- If the entity retrieved has the same name as the entity that follows it in the RACF database, ICHEINTY sets the duplicate data set name count. The count becomes 2 if it was zero on entry; otherwise, the count increases by one. The count is zero if the entity is not a duplicate of the one that follows it.
- ICHEINTY sets a return code of 44 (X'2C') if the values being returned do not fit into the provided work area, unless you specified WKSP which would provide you with an additional work area.
- For qualified types (data set, general, and connect), the located entity must have the same high-level qualifier as the specified entity. Otherwise, the macro sets a return code of 12 (X'0C').
- For data set profiles, the qualifier includes the first period in the name.
- For TYPE='DS', if the duplicate data set name count in the work area is not zero, ICHEINTY locates the specified data set name. (That is, if the duplicate data set count equals N, then the macro locates the Nth occurrence of the specified name. If there are less than N occurrences of the specified name, the same process occurs as when the duplicate data set count is zero; ICHEINTY locates the profile of the first entity of the specified type that follows the specified entity in the RACF data set.) If you want to locate the first DATASET profile with a name greater than or equal to the name specified by ENTRY= or ENTRYX, you can do so by setting the duplicate data set name count to one.
- If an ICHETEST macro specified in the TESTS operand failed, ICHEINTY sets a return code of 52 (X'34') and the NEXT operation terminates.
- If you specify a segment other than BASE on this ICHEINTY, or on any ICHEINTY in a chain, the RACF manager skips any profiles that do not contain an occurrence of the segment. Normal processing (TESTS on ICHEINTY, ICHEACTN) will resume with the next profile containing the segment. This simplifies the process of finding users who are defined to TSO, for example.

NEXTC
Is similar to the NEXT operation with the following exception:

- For qualified types (data set, general, and connect), ICHEINTY does not make the high-level qualifier check. For unqualified types (group and user), NEXTC processing is identical to NEXT processing. The qualifier for a general profile is the 8-character class name.

RENAME
Renames a data set, SFS file, or SFS directory entry in the RACF database.

You must specify the NEWNAME or NEWNAMX operand.

TYPE= 'GRP'|'USR'|'CON'|'DS'|'GEN'
Specifies the type of the entry as GROUP ('GRP'), USER ('USR'), CONNECT ('CON'), DATASET ('DS'), or general resource ('GEN').

The final parameter list the SVC uses as a request to the RACF manager must include a value for TYPE.

ENTRY= entry-address
Specifies the address of a 1-byte entry name length field followed by the entry name. The NEXT and NEXTC operations update this field. When using NEXT or NEXTC you should initialize the field in this way. To initialize the entry field, point to a field that has a length of 1 and a field of X'00'. The area pointed to must allow for 255 bytes of data to be returned.

The final parameter list the SVC uses as a request to the RACF manager must include a value for ENTRY or ENTRYX.

ENTRYX= extended-entry-address
Specifies the extended entry name field for long name support. You must specify the address of *two* 2-byte fields, followed by the entry name.

- The first 2-byte field specifies a buffer length which can be from 0 to 255 bytes. This length field only refers to the length of the buffer that contains the entity name; it does not include the length of either length field.
- The second 2-byte field specifies the actual length of the entity name. This length field includes only the length of the actual name without any trailing blanks; it does not include the length of either length field.

As with ENTRY, the NEXT and NEXTC operations update this field. The area pointed to must allow for a name which is of maximum entry length. ENTRY and ENTRYX are mutually exclusive keywords. It is recommended that you use the ENTRYX keyword because by allowing you to code to the specific amount of space that you need, you will save storage.

The final parameter list the SVC uses as a request to the RACF manager must include a value for ENTRY or ENTRYX. To use the ENTRYX keyword, you must specify RELEASE=1.9 or later.

CLASS= class-address
Specifies the address of an 8-character class name (left-justified and blank-padded, if necessary.) The class name is required when TYPE='GEN' and is ignored for all other types.

FLDACC =NO|YES
Specifies the presence or absence of field level access checking. If you specify FLDACC=YES, the RACF database manager checks to see that the user running your program has the authority to retrieve or modify the fields that have been specified in the ICHETEST and ICHEACTN macros associated with the current ICHEINTY macro.

Note: For field level access checking to occur, you must specify RELEASE=1.8 or later when you code the ICHEINTY and associated ICHETEST and/or ICHEACTN macros. RACF will bypass field access checking for any ICHETEST or ICHEACTN macro for which RELEASE=1.8 or later has not been specified. In addition, before your program executes, the security administrator must activate the FIELD class. If you code FLDACC=YES and the field class is not active, the request will be failed with a return code 60.

A further note: in addition, the security administrator must issue the RDEFINE and PERMIT commands to designate those users who will have the authority to access the fields designated in the ICHETEST and ICHEACTN macros.

If you specify FLDACC=NO or omit the parameter, the manager does not perform field level access checking.

RELEASE=number RELEASE=(,CHECK) RELEASE=(number,CHECK)
Specifies the release number. The release numbers you can specify with the ICHEINTY macro are 1.10, 1.9.2, 1.9, 1.8.1, 1.8, or 1.7.

Note: RELEASE=1.10 is only supported on the ICHEINTY macro. When an associated ICHETEST and/or ICHEACTN macro is involved with an ICHEINTY macro that has a RELEASE=1.10, the ICHETEST and/or ICHEACTN macro can and must specify a RELEASE=1.9.2.

When you specify 1.8 or later, the RACF manager returns data using the new 1.8 user work area format (documented under the sections entitled "Using ICHEACTN to Retrieve Data When the ICHEINTY Has DATAMAP=NEW" and "Using ICHEACTN to Retrieve Data When the ICHEINTY Has

DATAMAP=OLD" in this chapter). In effect, DATAMAP defaults to DATAMAP=NEW if you specify RELEASE=1.8 or later and omit DATAMAP.

If you specify RELEASE=1.7, or allow the release parameter to default to 1.7, the RACF manager returns data using the 1.7 user work area format. In this case, DATAMAP defaults to DATAMAP=OLD if you omit it.

If you want to use 1.8 parameters, and the 1.7 user work area format, you must specify RELEASE=1.8 or later and DATAMAP=OLD.

To use the 1.8 parameters, you must specify RELEASE=1.8 or later. If you specify RELEASE=1.8 or later the ICHEINTY parameter list must be in modifiable storage.

The default is RELEASE=1.7.

RUN =YES|NO

Specifies whether to activate or deactivate the parameter list. If you specify RUN=NO, the RACF manager ignores the request designated by this ICHEINTY macro although it will process the CHAIN parameter. If you specify RUN=YES, RACF processes this request and also processes the CHAIN parameter. Thus you can use the RUN parameter to deactivate or activate one or more ICHEINTY parameter lists without having to rearrange the chaining.

ACEE= acee-address

Specifies an ACEE that RACF uses to perform field authorization checking. If you specify FLDACC=YES, but omit ACEE, the RACF manager will use the appropriate ACEE pointed to either by the TCB or the ASXB.

WKSP= subpool-number

Specifies the number of a storage subpool. If the area specified by the WKAREA parameter is too small to contain the field data retrieved from LOCATE, NEXT, or NEXTC operation, the RACF manager obtains an additional workarea from this subpool in the caller's key. The RACF manager returns the address of the workarea in the fullword at offset 60 (X'3C') in the ICHEINTY parameter list. If additional storage is not needed, the RACF manager sets the fullword to zero. It is your responsibility to free any returned workarea; its subpool number is stored in the one byte field at offset 29 (X'1D') in the parameter list, its address in the fullword at offset 60 (X'3C') in the ICHEINTY parameter list and its size in the first fullword of the workarea itself.

Note:

1. Even if you specify WKSP, you must still provide a workarea at least 30 bytes long using the WKAREA operand.
2. You can simplify your coding if you use WKSP because you won't have to process a return code of 44 (X'2C').
3. If the RACF manager is unable to obtain a large enough workarea, an "out-of-storage" abend occurs.

CHAIN= parm-list address

Specifies a parameter list that another ICHEINTY macro created. This chained parameter list executes after the current one within the same manager invocation. If several ICHEINTY requests pertain to the same profile, you can use CHAIN to string the requests together. This chaining improves performance because the RACF manager retrieves the profile only once for the entire chain. Each ICHEINTY parameter list in the chain must contain the same values for the following parameters: ACEE, CLASS, ENTRY or ENTRYX, GENERIC, RBA, SMC, TYPE, INDEX, and VOLUME.

Note:

1. Because there are no connect profiles in the restructured database, chained ICHEINTY requests do not work as anticipated. Therefore, if you chain two ICHEINTY requests with TYPE=CON, the second ICHEINTY is ignored in the restructured database.
2. You cannot specify INDEX=ONLY or NEWNAME for any request in the chain.
3. When you chain ICHEINTY macros together, they must obey the following rules:

- The first ICHEINTY in the chain must be a LOCATE, NEXT/NEXTC, ALTER/ALTERI, ADD, or a DELETE with SEGMENT specified.
- The remaining ICHEINTY macros in the chain must be:
  - LOCATE, if the first was LOCATE
  - NEXT/NEXTC, if the first was NEXT or NEXTC
  - ALTERI, if the first was ALTERI
  - ALTER, if the first was ALTER, DELETE, or ADD
  - DELETE, with SEGMENT, if the first was ALTER or DELETE
- The RACF manager return code will be set to the highest return code of any of the individual ICHEINTY macros that have been chained together.

DATAMAP= OLD|NEW (default depends on RELEASE)
DATAMAP determines the format of the workarea returned for a LOCATE, NEXT, or NEXTC operation. You can specify the DATAMAP parameter in several different combinations to tailor it to your system:

- If you specify RELEASE=1.7 or allow RELEASE= to default to 1.7, you need not specify DATAMAP. It defaults to OLD, meaning the 1.7 format.

  Note: You cannot specify DATAMAP=NEW, if you specify RELEASE=1.7 or default to it.
- If you specify RELEASE=1.8 or later, and allow DATAMAP to default, it defaults to NEW, meaning the 1.8 format.
- If you specify RELEASE=1.8 or later, and want to use the 1.7 user work area format, you must specify DATAMAP=OLD for the data to be retrieved in the 1.7 format.

  Note: Releases prior to 1.7 are in the 1.7 format.

SEGMENT= 'segment name'
Specifies that this request is to apply to a specific segment in the profile. If you do not specify a specific segment, the default is the BASE segment. If you specify a segment other than the BASE segment, the operation cannot be ADD, DELETEA, or RENAME. If you specify a segment, then the DELETE operation will delete only that segment. If you do not specify a segment, then the DELETE operation will delete the entire profile. If you specify a segment, then the ALTER operation will alter only that segment. If you do not specify a segment then the ALTER operation will update the BASE segment. See Appendix F, "RACF Database Templates," on page 377 for a list of valid segment names.

VOLUME= volume-address
Specifies the address of a 6-character volume identifier. When TYPE='DS', the volume identifier differentiates among data sets with the same name. When the operation is ADD, TYPE='GEN', and the class is TAPEVOL, the volume identifier specifies the name of an existing tape volume set to which the current entry is to be added. In all other cases, ICHEINTY ignores the volume identifier.

ACTIONS= (action-address,.....)
Specifies the address of one or more ICHEACTN macros that determine which profile field(s) the RACF manager is to retrieve or update. See the description of the ICHEACTN macro later in this chapter. You can specify up to 255 actions.

Note: If you specify ACTION on the execute form of the ICHEINTY macro, the number of actions you specify should agree with the number of actions you have specified on the list form of the macro (or the FLDEF list, if you use that instead). If the numbers do not match, you must specify the OPTION keyword on the execute form to update the counts, using the appropriate ACTION operands.

TESTS= (test-addr[,[AND],test-addr]...)
Allows some preliminary testing on selected conditions prior to the execution of the operation specified by the ICHEINTY macro. You must specify an odd number of items (including the connector 'AND'). Each address must be the address of a list built by the ICHETEST macro. See "ICHETEST Macro" on page 324.

Note: If you specify TEST on the execute form of the ICHEINTY macro, the number of tests you specify should agree with the number of tests you have specified on the list form of the macro (or

the FLDEF list, if you use that instead). If the numbers do not match, you must specify the OPTION keyword on the execute form to update the counts, using the appropriate TESTC or TESTM operands.

WKAREA= wkarea-address
Specifies the address of the area into which the retrieved values are to be placed. This operand is valid and required only for the LOCATE, NEXT, and NEXTC operations. The workarea must be at least 30 bytes long.

User work area formats are described under "Using ICHEACTN to Retrieve Data When ICHEINTY Has DATAMAP=NEW" on page 329 and "Using ICHEACTN to Retrieve Data When the ICHEINTY Has DATAMAP=OLD" on page 332. For a related operand, see the previous description in this section of WKSP.

NEWNAME= newname-address
Specifies the address of the new name to be assigned to the entity named by the ENTRY operand. The name must be left-justified and followed by at least one blank.

Whereas ENTRY is a 1-byte length field followed by a name, NEWNAME specifies an entry name which is *not* preceded by a 1-byte length field. This operand is valid only for the RENAME operation and TYPE='DS'.

NEWNAMX= extended-newname-address
Specifies the address of the new name to be assigned to the entity in the ENTRYX keyword. The format of the new name is the same as that of the ENTRYX keyword.

- The first 2-byte field specifies a buffer length which can be from 0 to 255 bytes. This length field refers only to the length of the buffer that contains the entity name; it does not include the length of either length field.
- The second 2-byte field specifies the actual length of the entity name. This length field includes only the length of the actual name without any trailing blanks; it does not include the length of either length field.

NEWNAMX and NEWNAME are mutually exclusive parameters, as are NEWNAMX and WKAREA. The NEWNAMX keyword is valid only for the RENAME operation and for TYPE='DS'. To use NEWNAMX, you must specify RELEASE=1.9 or later.

RBA= RBA-address
Specifies the address of a 6-byte relative byte area (RBA) of a level one index that points to the profile to be altered. This keyword is valid only for an ALTERI request. RBA should specify the value returned by a previous LOCATE operation.

FLDEF= fldef-address
Specifies a remote list of ACTION/TEST pointers set up by an ICHEINTY with the FLDEF operation.

OPTIONS= ([NOPRO,TESTM,TESTC,ACTION,NOEXEC,FLDEF]...)
Provides more direct control of the code generated by the EXECUTE form of the macro. (This operand is valid only with the EXECUTE form of the macro.) You can specify one or any number of the following subfields:

NOPRO
Does not generate any prologue code; that is, the instructions that set the type of request, such as ADD, by updating the first two bytes of the parameter list, are not generated.

FLDEF
Generates the FLDEF pointer relocation code to point to the list of ACTION and TEST pointers in the ICHEINTY macro expansion.

ACTION
Generates code to set the number of ACTIONs that are to be performed.

TESTC
Generates code to set the number of TESTs that are to be performed.

TESTM
Generates code to set both actual and maximum number of TESTs.

NOEXEC
Does not generate the SVC instruction to invoke the RACF manager. This subfield is useful with the EXECUTE form of the macro to allow partial setup of the parameter list.

SMC= YES|NO
Controls the 'set-must-complete' operation mode of the RACF manager. YES is the default mode of operation.

GENERIC= NO|YES|UNCOND
Informs the RACF manager whether the given entity name is a generic name.

NO
Never generic.

The RACF manager does not attempt to convert the name specified by the ENTRY operand from external to internal form. GENERIC=NO is the default.

YES
May be generic.

The RACF manager attempts to convert the name specified by the ENTRY operand from external to internal form. The RACF manager does the conversion only if the entity name contains a generic character (an * or %). If the entity name does not contain a generic character, processing continues without any conversion.

UNCOND
Always generic.

The RACF manager unconditionally converts the name specified by the ENTRY operand from external to internal form.

For RENAME, the same process applies also to the NEWNAME operand.

MF= I|L|(E,address)
Specifies the form of the macro as either INLINE, LIST, or EXECUTE.

The INLINE form generates code to branch around the parameter list. In the MF=I form, the label names the instruction preceding of the parameter list. MF=I is the default.

The LIST form reserves and initializes storage.

The EXECUTE form modifies a list defined elsewhere. If you use the EXECUTE form, you must specify the address of the list to be modified. The address can be an A-type address or register (2 through 12).

Table 185. ICHEINTY Parameters

Parameter	RELEASE=1.7 and earlier	RELEASE=1.8 or later
ACEE		X
ACTIONS=	X	X
CHAIN=		X
CLASS=	X	X
DATAMAP=		X
ENTRY=	X	X
FLDACC		X
FLDEF=	X	X
GENERIC=	X	X

Table 185. ICHEINTY Parameters (continued)		
Parameter	**RELEASE=1.7 and earlier**	**RELEASE=1.8 or later**
INDEX=	X	X
MF=	X	X
NEWNAME=	X	X
OPTIONS=	X	X
RBA=	X	X
RELEASE=	X	X
RUN		X
SEGMENT		X
SMC=	X	X
TESTS=	X	X
TYPE=	X	X
VOLUME=	X	X
WKAREA=	X	X
WKSP		X

Return Codes from the ICHEINTY Macro

If you did not specify RELEASE=1.8 or later, Register 15 contains the ICHEINTY return code and Register 0 contains the reason code. If you specified RELEASE=1.8 or later, Register 15 contains the highest return code from any of the ICHEINTY macros; Register 0 contains the corresponding reason code. The return code for each ICHEINTY macro appears in the fullword at offset 52(X'34') in each ICHEINTY parameter list; the corresponding reason code appears in the fullword at offset 56(X'38').

Hex (Decimal)
Description

0 (0)
The requested operation was successful.

4 (4)
If the reason code is 0, a recovery environment could not be established; if the reason code is 4, an invalid function code was specified. (Valid functions are RACLIST, RACXTRT, and ICHEINTY. The parameter list was not valid for any of those functions.)

8 (8)
An attempt was made to add an entry to the RACF database but an identical entry already exists.

C (12)
For requests other than NEXT or NEXTC, the specified entry did not exist.

For NEXT or NEXTC requests, no subsequent entries satisfied the request.

10 (16)
Reserved.

14 (20)
The RACF database did not contain enough space to satisfy the request.

18 (24)
An I/O error occurred while accessing the RACF database.

1C (28)
RACF was not active at the time of the request.

20 (32)
The request type requires a user work area but the area was not provided (the address in the parameter list was 0) or for a RENAME, neither NEWNAME nor NEWNAMX was supplied.

24 (36)
The input parameter list or the associated ACTION and TEST blocks contain an error.

When this code is returned, the possible reason codes are:

1
Invalid entry name

2
Action specified with DELETE or DELETEA

3
An action or test specified for an undefined field

4
Test specified with RENAME

5
Reserved

6
Reserved

7
Incorrect entry type

8
GROUP=YES specified for an ICHEACTN, but the data length given was too long for the associated data. This reason code can occur with DATAMAP=OLD.

9
GROUP=YES specified for an ICHEACTN, but the data length given was too short for the associated data.

10
Chained ICHEINTY macros have inconsistent parameters: (CLASS, ENTRY, ENTRYX, GENERIC, RBA, SMC, TYPE, INDEX, or VOLUME).

11
Chained ICHEINTY macros have inconsistent request types (operations).

12
All ICHEINTY macros specified RUN=NO

13
Operation not allowed with SEGMENT keyword.

14
Illegal field specified for GROUP=YES, must be a repeat
group count field

15
More than 1000 ICHEINTY macros present in the chain

16
Specified SEGMENT name not allowed for the specified profile type

17
GROUP=YES specified for an ICHEACTN but the data length given was too long for the associated data. This reason code can occur with DATAMAP=NEW.

18
Data byte specified on ICHEACTN exceeded the length of the specified fixed-length field.

19
Inconsistency between action data length and repeat group fields. GROUP=YES data is too short.

20
Invalid ENTRYX. Current length is greater than 44 and either the primary or the backup database is not in the restructured format.

21
Invalid NEWNAMX. Current length is greater than 44, and either the primary or the backup database is not in the restructured (RDS) format.

22
Data length specified on the ICHEACTN macro was less than zero and neither FLDATA='DEL' nor FLDATA='COUNT' were specified.

23
The generic entity name exceeds the maximum length after it has been encoded.

26
Invalid date supplied on the ICHEACTN when DATEFMT=YYYYDDDF is specified. Date must be a length of 4 bytes and in the form CCYYDDDF where

- For cc=19, 70 < yy <= 99 and
- For cc=20, 00 <= yy <= 70.

28 (40)
The maximum profile size (65,535 bytes) has been reached; the profile cannot be expanded.

2C (44)
The user-supplied work area was not large enough to hold all the data returned. The work area is filled with data up to, but not including, the first field that did not fit. If WKSP was specified, the manager obtains a new workarea, retrieves the data, and sets the return code to 0.

30 (48)
The user-supplied work area was smaller than the minimum amount required (30 bytes).

34 (52)
A test condition specified in the TESTS keyword of the ICHEINTY macro was not met; further processing was suppressed.

38 (56)
You requested an operation on a DATASET type entry that has multiple RACF definitions, but you did not specify a VOLUME to single out a specific entry.

3C (60)
For DATASET type entries, you specified a VOLUME that did not exist in the volume list of any entry with the specified name. For TAPEVOL class entries, a request tried to add a new TAPEVOL to a nonexistent tape volume set.

40 (64)
You attempted to delete one of the IBM-defined entries (such as SYS1 or IBMUSER) from the RACF database.

44 (68)
An ALTERI request attempted to change the size of the profile being updated.

48 (72)
A request to add an entry to the RACF database would have caused the RACF index to increase to a depth that RACF does not support. The maximum depth is 10 levels.

4C (76)
ICHEINTY encountered an invalid index block or read a non-index block when it expected an index block.

50 (80)
You made an attempt to update (by a request other than ALTERI) a RACF database that has been extended (the 'extended' bit in the ICB was on).

54 (84)
Reserved.

58 (88)
At least one (but not all) ICHEACTN macros for information retrieval failed to be executed because of a profile field access violation.

5C (92)
All ICHEACTN macros for information retrieval failed to be executed because of a profile field access violation.

60 (96)
An ICHEACTN macro attempted to alter a field and failed because of a profile field access violation. All ICHEACTN macros for the ICHEINTY were suppressed. For FLDACC entries, the field class may not be active.

64 (100)
RELEASE=(1.8, 1.8.1, 1.9, CHECK) was specified on the E-Form ICHEINTY, but the L-form did not specify RELEASE=1.8 or later

68 (104)
The requested profile on the database contains erroneous data. A reason code is returned as follows:

1
The profile is physically too short to contain the data implied by variable field lengths or repeat group count fields.

74 (116)
The maximum length of extended entry of ICHEINTY parameter list is not enough to contain a found profile name.

88 (136)
Internal error during encryption of a field.

ICHETEST Macro

The ICHETEST macro tests for user-specified conditions on selected data in a RACF profile. You can use the ICHETEST macro with the ICHEINTY and/or ICHEACTN macros to ensure that a specific requirement is met before processing of the ICHEINTY and/or ICHEACTN macro occurs. Failure to meet the requirements specified on the ICHETEST macro causes further processing of the associated ICHEINTY or ICHEACTN macro to be suppressed.

The ICHETEST macro must be issued from a task running in non-cross-memory mode with no locks held. The issuing task must be authorized (APF-authorized, in system key 0-7, or running in supervisor state).

The format of the ICHETEST macro is:

```
[label] ICHETEST  FIELD=field-name|address
                  ,FLDATA=(length,address)
                  [,COND=EQ|NE|GT|LT|GE|LE|ONES|ZEROS|
                        MIXED]
                  [,ENCRYPT=TEMPLATE|YES|NO]
                  [,MF=L|(E,address)|I]
                  [,RELEASE=number|(,CHECK)|(number,CHECK)]
```

FIELD= field-name|address
Specifies the field-name in the RACF profile whose value is to be tested.

If you use the LIST form of the macro, specify the name of the field. The name must be from 1-to 8-characters long, not enclosed in quotes, and defined in the RACF template. In addition, the field cannot be a combination field name (such as ACL in the group profile). Note, however, that a combination field that specifies only one associated field is allowable. Such a combination field is called an alias field such as OWNER in the GROUP profile.

If you use the EXECUTE or INLINE form of the macro, specify the address of the field name to be tested. The address can be an A-type address or register (2 through 12). For EXECUTE and INLINE, you can also specify the field name as a constant (for example, ‘OWNER’).

FLDATA= (length,address)
Specifies the data to be tested against.

The length must be greater than zero and less than or equal to the length of field-name in the FIELD operand, or the test will fail. For fixed length fields, you can specify a length that is less than the actual length of the field in the profile. For flag fields, the length specified is ignored and a 1-byte length is assumed. For variable-length fields, if the length is not equal to the field length in the profile, the test fails unless COND=NE is specified. Also, for variable-length fields the field data must not contain a length byte.

COND= EQ|NE|GT|LT|GE|LE|ONES|ZEROS|MIXED
Specifies the relationship that must exist between the FLDATA and FIELD values to satisfy the test, for example, (FLDATA value) GE FIELD (value FIELD).

EQ, NE, GT, LT, GE, and LE are valid only for fixed length or variable-length fields. They are invalid for flag fields.

ONES, ZEROS, and MIXED are valid only for flag fields.

If you omit this operand, COND=EQ is the default. An explanation of ONES, ZEROS, and MIXED follows:

- **ONES**—If the 1 bits exist in the FIELD value base where the 1 bits exist in the FLDATA value,then the test is successful.
- **ZEROS**—If the 0 bits exist in the FIELD value where the 1 bits exist in the FLDATA value,then the test is successful.
- **MIXED**—If both 0 bits and 1 bits exist in the FIELD value where 1 bits exist in the FLDATA value, then the test is successful.

You can think of this operation as an AND operation between the database and the test field, followed by a branch on the condition code.

ENCRYPT= TEMPLATE|YES|NO
Specifies whether the data specified by FLDATA is to be encoded before the test is performed. If ENCRYPT=YES, the data is encoded regardless of whether the template flag associated with the field specifies that it is to be encoded. If ENCRYPT=NO, RACF does not encode the data regardless of the template flag value. If ENCRYPT=TEMPLATE, the template flag determines whether the data is encoded.

ENCRYPT is ignored if you specify COND as ONES, ZEROS, or MIXED.

MF= L|(E,address)|I
Specifies the form of the macro as either LIST, EXECUTE, or INLINE.

The LIST form reserves and initializes storage. MF=L is the default.

The EXECUTE form modifies a list defined elsewhere. If you use the EXECUTE form, you must specify the address of the list to be modified. The address can be an A-type address or register (2 through 12).

The INLINE form is similar to a STANDARD form, except that it generates code to branch around the parameter list. In the MF=I form, the label names the first location of the parameter list, not the preceding instruction.

RELEASE=*number* RELEASE=(,CHECK) RELEASE=(*number*,CHECK)
Specifies the release number. The release numbers you can specify with the ICHETEST macro are 1.9.2, 1.9, 1.8.1, 1.8, or 1.7.

Note: RELEASE=1.10 is not supported on the ICHETEST macro. When an associated ICHETEST and/or ICHEACTN macro is invoked with an ICHEINTY macro that has a RELEASE=1.10, the ICHETEST and/or ICHEACTN macro can and must specify a RELEASE=1.9.2

Table 186. ICHETEST Parameters		
Parameter	**RELEASE=1.7 and earlier**	**RELEASE=1.8 or later**
COND=	X	X
ENCRYPT=	X	X
FIELD	X	X
FLDATA	X	X
MF	X	X

Some considerations regarding the ICHETEST macros are:

- You cannot use the ICHETEST macro with an ICHEINTY macro that has the RENAME operation specified.
- A profile can contain repeat groups. A repeat group consists of one or more sequential fields that can be repeated in the profile. By specifying COND=EQ, you can select the occurrence of the repeat group to which the action applies.

  By specifying COND=NE, you can position yourself past the last occurrence of the repeat group. Then you can add a new occurrence to the end of that repeat group with an ICHEACTN macro.

  Note: When the ICHEACTN macro refers to a repeat group and more than one ICHETEST macro is specified, the last ICHETEST macro serves to position data retrieval from the profile. Therefore, the last ICHETEST should refer to the same repeat group as the last ICHEACTN; otherwise the retrieved data will be from the last tested field. On multiple tests with fields in repeat groups, each test is processed separately, and if all succeed, the tests are considered to have succeeded.
- Tests involving negative numbers cause unpredictable results.
- If a specified address equals zero, ICHETEST makes no test.
- Use only COND=EQ or COND=NE to test masked fields. Other comparisons cause unpredictable results.
- The expansion of the ICHETEST macro MF=L or MF=I includes at offset 1, a 1-byte field whose value will be X'00' if the test was successful, or X'01' if the test failed. The ICHETEST parameter list must be in modifiable storage.
- If RELEASE=1.8 or later, the expansion of the ICHETEST macro MF=L or MF=I includes a one byte field at offset 3 whose low-order bit will be set to X'01' if the test failed because FLDACC=YES was specified on the associated ICHEINTY.
- It is possible to mix 1.7 and 1.8 or later format tests in the same request. The ICHEINTY and ICHEACTN macros can specify either RELEASE=1.7 or RELEASE=1.8 or later.
- When ICHEINTY LOCATE is used to retrieve data from a profile segment other than the BASE segment, default values (binary zeros for fixed-length fields, lengths of zero for variable length fields) will be returned by the manager if the profile could contain but does not contain an occurrence of that segment. If you need to know whether the segment actually exists, specify a TEST for the SEGNAME on the ICHEINTY. For example, when doing a LOCATE to retrieve the TSO segment from a user profile, use TEST as follows:

```
    ICHETEST  FIELD=SEGNAME,COND=EQ,FLDATA=(8,CTSO)
    ....
    ....
CTSO  DC  CL8'TSO'
```

ICHEACTN Macro

You can use the ICHEACTN macro together with the ICHEINTY to retrieve or alter data in a specified RACF profile. ICHEACTN builds a parameter list containing the RACF profile field name and, optionally, the addresses of ICHETEST macros that control the data processing.

The ICHEACTN macro must be issued from a task running in non-cross-memory mode with no locks held. The issuing task must be authorized (APF-authorized, in system key 0-7, or running in supervisor state).

The format of the ICHEACTN macro is:

```
[label] ICHEACTN  FIELD=field-name|address
                  ,FLDATA=(length,address)|'DEL'|'COUNT'
                  [,TESTS=(address[,AND,address]...)]
                  [,RUN=YES|NO]
                  [,GROUP=YES|NO]
                  [,ENCRYPT=TEMPLATE|YES|NO]
                  [,MF=L|(E,address)|I]
                  [,RELEASE=number|(,CHECK)|(number,CHECK)]
```

FIELD=field-name|address

Specifies the field-name in the RACF profile whose value is to be retrieved or updated. The field must be one that is defined in the RACF database template.

Do not specify FIELD to be the first field in a database segment because the user cannot retrieve or update the first field in a segment. In the restructured database templates, this field has a field ID of 001, and is usually described in the '**Field Being Described**' column as 'Start of Segment Fields'.

If you use the LIST form of the macro, specify the name of the field. The name must be 1-to 8-characters long, and not enclosed in quotes.

If you use the EXECUTE or INLINE form of the macro, specify the address of the name of the field to be retrieved or updated. The address can be an A-type address or register (2 through 12). For EXECUTE and INLINE, you can also specify the field name as a constant (for example, 'OWNER').

FLDATA= (length,address)|'DEL'|'COUNT'

Updates or deletes data in a specified RACF profile. This operand is valid when used with the ALTER, ALTERI, ADD and RENAME operations on the ICHEINTY macro. It is also valid with LOCATE, NEXT or NEXTC if RELEASE=1.8 or later. The ICHEACTN macro will have eight bytes reserved to hold the length and address of the retrieved data. In no case will a LOCATE, NEXT, or NEXTC return data into a field whose address is given in the ICHEACTN macro.

When you use ICHEACTN to replace modify data, the address points to a field which contains the value that is to replace the data in the specified FIELD of the profile. The address may be an A-type address or general register ((2) through (12)). The length specifies the size of the replacement field, and must be an integer constant or register ((2) through (12)).

When you use ICHEACTN to retrieve data and you specify RELEASE=1.8 or later, the RACF manager places the size of the retrieved field in the word at offset 12(X'0C') and the address of the data in the word at offset 16(X'10') of the ICHEACTN parameter list if no tests are specified. The addresses specified in TESTS= are placed before the FLDATA entries within the parameter list. Therefore, for each address noted within TESTS=, the FLDATA entries are displaced by four bytes. The use of the TESTS= operand increments these offsets by four bytes for each test specified regardless of whether DATAMAP=NEW or DATAMAP=OLD is specified.

'DEL' causes field-name in the FIELD operand to be given a null value or causes an occurrence of a repeat group to be deleted, or (if GROUP=YES is coded) deletes all occurrences of a repeat group.

'COUNT' causes field-name in the FIELD operand to be treated as a positive integer and increased by one.

When replacing or adding data, the length and address are processed as follows:

- If DATAMAP=OLD is specified or defaulted on the ICHEINTY:

If length is 0 or omitted, or the address is 0 or omitted, the specified field will be given a null value (a variable-length field is set to a length of 0; a flag field is set to X'00'; other fixed-length fields are set to all ‘FF’).

If the length is greater than 0, and the address is specified, the length will be ignored. The length of a fixed-length field is taken from the template, and the length of a variable length field is taken from the first byte of the data.

- If DATAMAP=NEW is specified on the ICHEINTY:

  If the length is 0 or omitted, or the address is 0 or omitted, the field is given a null value as indicated above. Otherwise, the field is set from the data specified, with the length specified. For a fixed-length field, if the specified length is less than the length given in the template, the value will be left-adjusted and filled with X'00's to the template length. If the length is greater than the template length, the operation will fail. For variable-length fields, the specified length is used; the first byte of the data is not used as the data length, but rather is considered to be data.

TESTS= (address[,[AND],address]...)
Specifies preliminary testing that must occur before any data retrieval or updating takes place. Each address specified must be the address of a list built by an ICHETEST macro. The address can be an A-type address or register (2 through 12). Multiple addresses indicate that all conditions (tests) must be satisfied. If not, RACF suppresses further processing of the macro. If you omit the logical connector ‘AND’, you must use a comma to indicate its omission.

Note: If GROUP=YES is also coded on the ICHEACTN macro, all tests specified by the TESTS parameter are ignored unless RELEASE=1.8 or later is also specified.

The addresses specified in TESTS= are placed before the FLDATA entries within the parameter list. Therefore, for each address noted within TESTS=, the FLDATA entries are displaced by four bytes. The use of the TESTS= operand increments these offsets by four bytes for each test specified regardless of whether DATAMAP=NEW or DATAMAP=OLD is specified.

RUN= YES|NO
Specifies if a data retrieval or update is to be actually performed. This operand allows you to code an ACTION operand on the ICHEINTY macro without the action being performed for this particular execution. The default is RUN=YES.

GROUP= YES|NO
Specifies whether an update for a repeat group is for a single occurrence of the group or for the entire group, including the repeat count that contains the number of occurrences. If FIELD=field-name contains the name of a repeat group count field and GROUP=YES, ICHEACTN replaces or deletes the entire repeat group, including the count field. The data format used with GROUP=YES depends on the DATAMAP value on the ICHEINTY. See “Using ICHEACTN to Alter Data When the ICHEINTY Has DATAMAP=NEW” on page 332 and “Using ICHEACTN to Alter Data When ICHEINTY Has DATAMAP=OLD” on page 334 for details.

Note: If GROUP=YES is also coded on the ICHEACTN macro all tests specified by the TESTS parameter are ignored unless RELEASE=1.8 or later is specified.

ENCRYPT= TEMPLATE|YES|NO
Specifies whether the data specified by FLDATA is to be encoded. If ENCRYPT=YES, the data is encoded regardless of whether the template flag associated with the field specifies that it is to be encoded. If ENCRYPT=NO, RACF does not encode the data regardless of the template flag value. If ENCRYPT=TEMPLATE, the template flag determines whether the data is encoded.

MF= L|(E,address)|I
Specifies the form of the macro as either LIST, EXECUTE or INLINE.

The LIST form reserves and initializes storage. MF=L is the default. If RELEASE=1.8 or later is specified, the storage must be modifiable, that is, not within a re-entrant module.

The EXECUTE form modifies a list defined elsewhere. If you use the EXECUTE form, you must specify the address of the list to be modified. The address can be an A-type address or register (2 through 12).

The INLINE form is similar to a STANDARD form, except that it generates code to branch around the parameter list. In the MF=I form, the label names the first location of the parameter list, not the preceding instruction.

RELEASE=*number* RELEASE=(,CHECK) RELEASE=(*number*,CHECK)
Specifies the release number.

The release numbers you can specify with the ICHEACTN macro are 1.9.2, 1.9, 1.8.1, 1.8, or 1.7.

Note: RELEASE=1.10 is not supported on the ICHEACTN macro. When an associated ICHETEST and/or ICHEACTN macro is invoked with an ICHEINTY macro that has a RELEASE=1.10, the ICHETEST and/or ICHEACTN macro can and must specify a RELEASE=1.9.2

When you specify 1.8 or later, the RACF manager returns data using the new 1.8 user work area format (documented under the section entitled "Using ICHEACTN to Retrieve Data" in this chapter). In effect, DATAMAP defaults to DATAMAP=NEW, if you specify RELEASE=1.8 or later and omit DATAMAP.

If you specify RELEASE=1.7 or allow the release parameter to default to 1.7, the RACF manager returns data using the 1.7 user work area format. In this case, DATAMAP defaults to DATAMAP=OLD if you omit it.

If you want to use 1.8 parameters, and the 1.7 user work area format, you must specify RELEASE=1.8 or later and DATAMAP=OLD.

To use the 1.8 parameters, you must specify RELEASE=1.8 or later. If you specify RELEASE=1.8 or later, the ICHEINTY parameter list must be in modifiable storage. The parameter list will include at offset 3 a byte whose low order bit (X'01') will be set if the action failed because of field level access checking.

The default is RELEASE=1.7.

Table 187. ICHEACTN Parameters

Parameter	RELEASE=1.7 and earlier	RELEASE=1.8 or later
ENCRYPT=	X	X
FIELD=	X	X
FLDATA=	X	X
GROUP=	X	X
MF=	X	X
RUN=	X	X
TESTS=	X	X

Using ICHEACTN With the DATAMAP=NEW and DATAMAP=OLD Operands

With Release 1.8, customers can choose between using their old datamap format and the new 1.8 datamap format. The following sections explain the relationship between the DATAMAP keyword and the RELEASE keyword. In addition, this section explains how to use the ICHEACTN to retrieve and alter data when the ICHEINTY macro has DATAMAP=NEW specified and how to use ICHEACTN when the ICHEINTY macro has DATAMAP=OLD specified.

Using ICHEACTN to Retrieve Data When ICHEINTY Has DATAMAP=NEW

The ICHEACTN macro retrieves data when used with the ICHEINTY macro which has a LOCATE, NEXT or NEXTC operand. With DATAMAP=NEW on the ICHEINTY and RELEASE=1.8 or later on the ICHEACTN, data retrieval and modification are compatible operations. That is, you can do an ICHEINTY LOCATE followed by an ICHEINTY ALTER (with the same ICHEACTN) and the profile will end up with its original

data. Or alternatively, by changing the ENTRY name you could copy data from one profile to another. When using ICHEACTN to retrieve data, you must supply a work area on the ICHEINTY macro into which the retrieved data can be placed. The first fullword of the work area must be the length of the work area (including the first fullword itself). The minimum work area is 30 bytes, even if no data is being retrieved.

The format of the user work area is as follows:

Offset (hex) Length Description
0 4 Length of entire work area
4 6 RBA return area
A 1 Flags
B 1 Reserved
C 4 Duplicate data set name count
10 8 Reserved
18 4 Length of data returned into work area
1C variable Field value return area

Ensure that the storage in the work area from +4 to +1E is initialized to binary zeros. If the area is not initialized, it can be difficult to determine if the information returned by the RACF manager is present.

If the profile located has a generic name, bit 0 (X'80') of the flag byte at offset (X'0A') is set to on.

An ICHEINTY macro can have several ICHEACTN macros associated with it. For each ICHEACTN macro, the RACF manager returns into the field value return area:

- A 4-byte length field. This length field contains the length of the retrieved data for that particular ICHEACTN macro. Note that this 4-byte length field does not contain its own length.
- The retrieved data from the RACF profile.
  - Simple variable-length fields are not preceded by an additional length byte as in the old format.
  - Within a combination field, each field is preceded by its respective four byte length field.
  - An alias field (combination field made up of only one field) does not have an extra length field.
  - Repeat group count fields are four bytes long, not two.
  - When replacing or retrieving an entire repeat group using (GROUP=YES), the repeat group count field does not precede the data.

When multiple ICHEACTNs are used, each returns data immediately following the data (if any) returned by the preceding ICHEACTN.

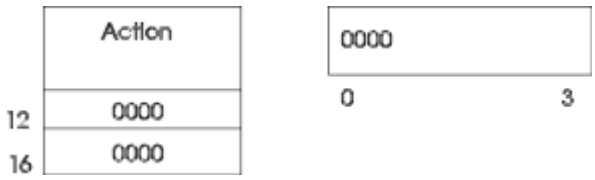
Note that all the fields are byte-aligned. In addition, if the ICHEACTN contains RELEASE=1.8 or later, the manager places the data length in the fullword at offset 12(X'0C') of the ICHEACTN and places a pointer to the data in the fullword at offset 16(X'10') of the ICHEACTN parameter list if no tests are specified. You must increment these offsets by 4 for each test specified by the ICHEACTN TESTS= parameter.

For example, with two tests, the length is returned at X'14' and the address is returned at X'18'. The addresses specified in TESTS= are placed before the FLDATA entries within the parameter list. Therefore, for each address noted within TESTS=, the FLDATA entries are displaced by four bytes.

The use of the TESTS= operand increments these offsets by four bytes for each test specified regardless of whether DATAMAP=NEW or DATAMAP=OLD is specified. The following examples show both the format of the returned data and the values that would be placed in the ICHEACTN if you specify RELEASE=1.8 or later.

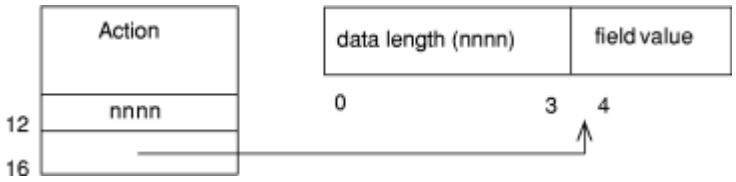
Some examples of the different field types that the RACF manager can return in the field value return area are:

1. If a condition specified by an ICHETEST macro (that is associated with the ICHEACTN macro) was not satisfied or if the specified field was a repeat field that contained no members, or if the action was failed by field level access checking, the field value area will not be returned and the length area will be equal to X'00000000'.



2. If the field specified is a fixed-length field, a variable-length field, a flag field, or a repeat group count field (GROUP=NO), the return field contains the length of the field followed by the field value.

   Note: A flag field is always one byte long. A repeat group count field is always four bytes long if GROUP=NO. An alias field is processed the same way as the simple field of which it is an alias.

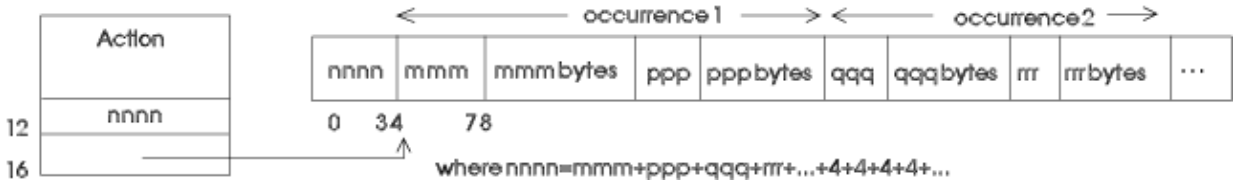


3. If the field specified is a combination field, the return area contains the length of all the fields in the combination, followed by a concatenation of the individual simple fields in the combination. Each simple field is returned as described above in (2).

   For example, if the combination contains two simple fields:

4. If the field specified is a field in a repeat group or a combination field made up of one or more fields in the same repeat group, the results returned depend upon whether (1) an ICHETEST macro was associated with the ICHEACTN in order to position to a particular occurrence of the repeat group or (2) no ICHETEST macro was associated and all occurrences are implied.

   When an ICHETEST is associated, the format of the result is the same as if the field were not in a repeat group. When no ICHETEST is associated, the result is the four-byte length field followed by the concatenation of the values of every occurrence of the specified field in the format shown above. If the specified field is a combination field, the values of the fields in the combination are first concatenated for each occurrence, then these concatenations are concatenated in the order of their occurrence.



5. If the field is a repeat-group count field, and the ICHEACTN specifies GROUP=YES, then the retrieved data contains all occurrences of the repeat group, in the following format:

   Where nnnn is the total length of data returned, mmmm is the length of occurrence 1, and ppp is the length of occurrence 2.

   Each occurrence will be formatted as though it were a combination field (see Note 3 page "3" on page 331) of all template fields defined for the group. For example, data set profiles have a field called

ACLCNT; the fields in the group are USERID, USERACS, and ACSCNT. An ICHEACTN to retrieve ACLCNT, with GROUP=YES, would return the following data if ACLCNT has the value 2:

```
nnnn (length of data)               DC AL4(54)
mmmm (length of occurrence 1)       DC AL4(23)

Declares for occurrence 1           DC AL4(8)
                                    DC CL8 'userid1'
                                    DC AL4(1)
                                    DC AL1(useracs1)
                                    DC AL4(2)
                                    DC AL2(acscnt1)

pppp (length of occurrence 2)       DC AL4(23)

Declares for occurrence 2           DC AL4(8)
                                    DC CL8'userid2'
                                    DC AL4(1)
                                    DC AL2(useracs2)
                                    DC AL4(2)
                                    DC AL2(acscnt2)
```

Using ICHEACTN to Alter Data When the ICHEINTY Has DATAMAP=NEW

The ICHEACTN macro alters data when used with the ICHEINTY macro having an ADD, ALTER, ALTERI, or RENAME operand. If the conditions specified by the TESTS keyword on the ICHEACTN macro are met, the field specified in the FIELD operand is assigned the value specified in the FLDATA operand. If the specified field in the RACF profile is in a repeat group, then:

- If you specified a test with COND=EQ, the existing occurrence of the repeat group is altered.
- If you specified a test with COND=NE, a new occurrence is added to the end of the repeat group.
- If you did not specify a test, a new occurrence is added to the beginning of the repeat group.

When replacing data, the FLDATA parameter should describe the size of the data and its address in the same format as shown above for retrieving data. When specifying a combination field, the total size must equal the sum of the individual sizes, including the length fields or the request fails.

The specification of FLDATA='COUNT' causes the specified fields to be treated as a positive integer and increased by one. If the field specified is variable length or has a fixed length greater than four, RACF ignores the specification and does not modify the field value.

If you specify FLDATA='DEL', the specified field has a null value; that is:

- For a fixed-length field that is not in a repeat group, the field is set to binary ones.
- For a flag field that is not in a repeat group, the field is set to binary zeros.
- For variable-length fields that are not in a repeat group, the length of the field is set to zero.
- For fields within a repeat group, the entire occurrence is deleted.

If you specify zero as the "address" value, the result is the same as if you had specified FLDATA='DEL', except that for fields in a repeat group, the field in the occurrence is set to a null value (the same as fields not in a repeat group).

If you specify FLDATA='DEL' or FLDATA='COUNT' on an ICHEACTN, the length field of the ICHEACTN is set to -1 or -2. If you also specify RELEASE=1.8 or later, and subsequently use the ICHEACTN to retrieve data, these new values will be lost. To avoid this, you should not use the same ICHEACTN for both DEL/COUNT and retrieval processing; or you should use the E-Form to re-establish DEL/COUNT after the data retrieval.

Using ICHEACTN to Retrieve Data When the ICHEINTY Has DATAMAP=OLD

The ICHEACTN macro retrieves data when used with the ICHEINTY macro having a LOCATE, NEXT or NEXTC operand. When using ICHEACTN to retrieve data, you must supply a work area on the ICHEINTY macro into which the retrieved data can be placed. The first fullword of the work area must be the length of the work area (including the first fullword itself). The minimum work area is 30 bytes, even if no data is being retrieved.

The format of the user work area is as follows:

Offset (hex)	Length	Description
0	4	Length of entire work area
4	6	RBA return area
A	1	Flags
B	1	Reserved
C	4	Duplicate data set name count
10	8	Reserved
18	4	Length of data returned into work area
1C	variable	Field value return area

Ensure that the storage in the work area from +4 to +1E is initialized to binary zeros. If the area is not initialized, it can be difficult to determine if the information returned by the RACF manager is present.

If the profile located has a generic name, bit 0 (X'80') of the flag byte at offset (X'0A') is on. This flag bit is useful when performing NEXT or NEXTC operations to process many profiles.

An ICHEINTY macro can have several ICHEACTN macros associated with it. For each ICHEACTN macro, the RACF manager returns into the field value return area:

- A 2-byte length field. This length field contains the length of the retrieved data for that particular ICHEACTN macro. Note that this 2-byte length field does not contain its own length.
- The retrieved data from the RACF profile.

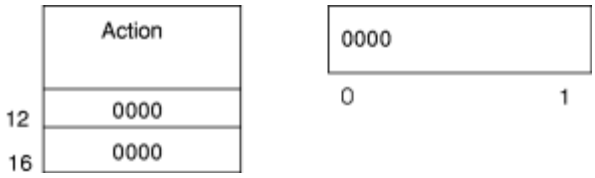
Note that all the fields are byte-aligned. In addition, if the ICHEACTN contains RELEASE=1.8 or later, the manager will place the data length in the fullword at offset 12(X'0C') of the ICHEACTN, and will place a pointer to the data in the fullword at offset 16(X'10') of the ICHEACTN parameter list if no tests are specified. You must increment these offsets by 4 for each test specified by the ICHEACTN TESTS= parameter.

For example, with two tests, the length is returned at X'14' and the address is returned at X'18'. The addresses specified in TESTS= are placed before the FLDATA entries within the parameter list. Therefore, for each address noted within TESTS=, the FLDATA entries are displaced by four bytes. The use of the TESTS= operand increments these offsets by four bytes for each test specified regardless of whether DATAMAP=NEW or DATAMAP=OLD is specified.

The following examples show both the format of the returned data and the values that would be placed in the ICHEACTN if you specify RELEASE=1.8 or later.

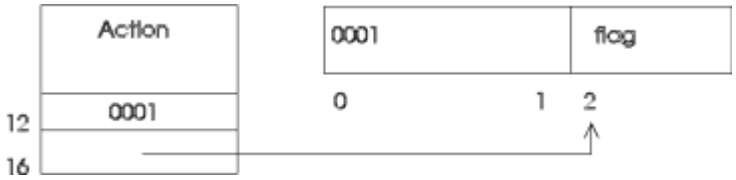
Some examples of the different field types that the RACF manager can return in the field value return area are:

1. If a condition specified by an ICHETEST macro (that is associated with the ICHEACTN macro) was not satisfied or if the specified field was a repeat field that contained no members, the field value area will not be returned and the length area will be equal to X'0000'.

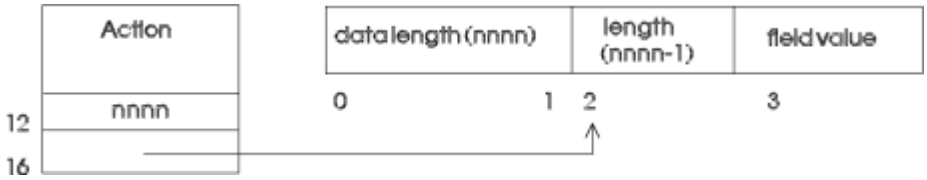


2. If the field specified is a fixed-length field, the return field contains the length of the field followed by the field value.

3. If the field specified is a flag field, the return field contains the length of the field (X'0001') followed by a 1-byte value.



4. If the field specified is a variable-length field, the return field contains the length of the field followed by a 1-byte length field (that does not include its own length) followed by the field value.



5. If the field specified is a combination field, the return area contains the length of all the fields in the combination, followed by a concatenation of values of each of the individual fields in the combination. If a field in the combination is in a repeat group, all the fields in the combination must be in the same repeat group. (Example 6 shows how the RACF manager returns combinations containing fields of a repeat group.)

6. If the field specified is a field in a repeat group or a combination field made up of one or more fields in the same repeat group, the results returned depend on whether (1) an ICHETEST macro was associated with the ICHEACTN in order to position to a particular occurrence of the repeat group or (2) no ICHETEST macro was associated and all occurrences are implied.

   When an ICHETEST is associated, the format of the result is the same as if the field were not in a repeat group. When no ICHETEST is associated, the result is the two-byte length field followed by the concatenation of the values of every occurrence of the specified field. If the specified field is a combination field, the values of the fields in the combination are first concatenated for each occurrence, then these concatenations are concatenated in the order of their occurrence.

Using ICHEACTN to Alter Data When ICHEINTY Has DATAMAP=OLD

The ICHEACTN macro alters data when used with the ICHEINTY macro having an ADD, ALTER, ALTERI, or RENAME operand. If the conditions specified by the TESTS keyword on the ICHEACTN macro are met, the field specified in the FIELD operand is assigned the value specified in the FLDATA operand. If the specified field in the RACF profile is in a repeat group, then:

- If you specified a test with COND=EQ, the existing occurrence of the repeat group is altered.
- If you specified a test with COND=NE, a new occurrence is added to the end of the repeat group.
- If you did not specify a test, a new occurrence is added to the beginning of the repeat group.

RACF uses the length specified as a subfield of the FLDATA keyword only when you specify GROUP=YES. For fixed-length fields, the data length is the field length in the template. For variable-length fields, the data length is the first data byte (it does not include its own length). RACF handles combination fields as a succession of fields, either fixed or variable length. If the combination field contains some but not all of the fields in a repeat group, the fields not included are set to null values.

The specification of FLDATA='COUNT' causes the specified field to be treated as a positive integer and increased by one. If the field specified is variable length or has a fixed length greater than four, RACF ignores the specification and does not modify the field value.

If you specify FLDATA='DEL', the specified field is given a null value; that is :

- For a fixed-length field that is not in a repeat group, the field is set to binary ones.
- For a flag field that is not in a repeat group, the field is set to binary zeros.
- For variable-length fields that are not in a repeat group, the length of the field is set to zero.
- For fields within a repeat group, the entire occurrence is deleted.

If you specify zero as the "address" value, the result is the same as if you had specified FLDATA='DEL', except that for fields in a repeat group, the field in the occurrence is set to a null value (the same as fields not in a repeat group).

Examples of ICHEINTY, ICHETEST, and ICHEACTN Macro Usage

The following examples illustrate some of the functions provided by the ICHEINTY, ICHETEST, and ICHEACTN macros:

1. Determining if a user is defined to RACF

```
*        .
*        .
*        .
         LA    15,WEND-W     LENGTH OF WORK AREA.
         ST    15,W          INITIALIZE WORK AREA.
         XC    WR,WR         CLEAR RESERVED AREA.
         ICHEINTY LOCATE,TYPE='USR',ENTRY=USR1,WKAREA=W
         LTR   15,15         R15=0 IF USER DEFINED TO
                             RACF
         BNZ   NOTDEFD
*        .
*        .
*        .
*    DATA AREAS
USR1     DS    AL1           LENGTH OF USERID (1 TO 8)
         DS    CL8           USERID
W        DS    0F
         DS    F             LENGTH OF WORK AREA.
WR       DS    CL24          RESERVED.
         DS    F
WEND     EQU   *             END OF WORK AREA.
```

The ICHEINTY macro identifies the user profile to be located. A return code of 0 (X'00') in register 15 indicates that the user is defined to RACF. A return code of 12 (X'0C') indicates that the user is not defined. Note that this ICHEINTY macro contains a work area. By also coding an ICHEACTN macro in this example, you can retrieve current field values from this user profile into the work area.

2. Adding a user ID to a data set access list

```
*        .
*        .
*        .
         ICHEINTY ALTER,TYPE='DS',ENTRY=DSN1,            *
               ACTIONS=AACL
         LTR   15,15             0 RETURNED IF DS IS RACF
                                 DEFINED
         BNZ   DSNOTDEF          DS NOT RACF DEFINED OR
                                 ERROR
         CLI   TUSERID+1,X'00'   WAS USER ALREADY IN LIST
         BNZ   INLIST            YES.  USER WAS IN LIST
                                 ALREADY
*        .
*        .
*        .
*     DATA AREA
AACL     ICHEACTN FIELD=ACL,FLDATA=(11,ACL),             *
               TESTS=TUSERID,MF=L
TUSERID  ICHETEST FIELD=USERID,FLDATA=(8,USER),COND=NE,  *
               MF=L
```

```
DSN1     DS    AL1               DATA SET NAME LENGTH
                                 (1 TO 44)
         DS    CL44              DATA SET NAME
ACL      DS    0CL11             ACCESS LIST ENTRY
USER     DS    CL8               USERID TO BE ADDED
USERACS  DS    XL1               ACCESS TO BE GIVEN:
*                                X'80' FOR ALTER
*                                X'40' FOR CONTROL
*                                X'20' FOR UPDATE
*                                X'10' FOR READ
*                                X'01' FOR NONE
ACSCNT   DC    XL2'0000'         ZERO ACCESS COUNT
```

The ICHEINTY macro identifies the data set profile whose access list is to be updated. It also points to an ICHEACTN macro that describes how the profile is to be updated. In this example, RACF adds a user ID to the access list.

The ICHEACTN macro, in turn, points to an ICHETEST macro that tests for certain conditions before the profile can be updated. In this example, ICHETEST tests to determine if the specified user ID already exists in the access list. (The second byte of the test block at TUSERID is 0 if the user ID is not in the access list.) If the user ID does not exist, RACF adds the user ID (with the specified access authority) to the access list and updates the data set profile. If the user ID already exists, no profile update occurs.

3. Changing the access authority of a user in a data set access list

```
*        .
*        .
*        .
         ICHEINTY ALTER,TYPE='DS',ENTRY=DSN1,            *
               ACTIONS=AUSRACS
         LTR   15,15             0 RETURNED IF DS IS RACF
                                 DEFINED
         BNZ   DSNOTDEF          DS NOT RACF DEFINED OR
                                 ERROR
         CLI   TUSERID+1,X'00'   WAS USER IN LIST
         BNZ   NOTINLST          NO.  USER WAS NOT IN
                                 LIST
*        .
*        .
*        .
*     DATA AREA
AUSRACS  ICHEACTN FIELD=USERACS,FLDATA=(1,USERACS),      *
               TESTS=TUSERID,MF=L
TUSERID  ICHETEST FIELD=USERID,FLDATA=(8,USER),COND=EQ,  *
               MF=L
DSN1     DS    AL1               DATA SET NAME LENGTH
                                 (1 TO 44)
         DS    CL44              DATA SET NAME
UACC     DS    XL1               ACCESS TO BE GIVEN:
*                                X'80' FOR ALTER
*                                X'40' FOR CONTROL
                                 X'20' FOR UPDATE
                                 X'10' FOR READ
                                 X'01' FOR NONE
```

This example is similar to the previous example. However, if the user ID exists in the data set access list, RACF changes that user's access authority to the value specified in USERACS and updates the data set profile. If the user ID does not exist, no profile update occurs.

Note that you can use this example to delete a user ID from the data set access list by changing the ICHEACTN macro to read:

```
AUSRACS  ICHEACTN FIELD=USERID,FLDATA='DEL',             *
               TEST=TUSERID,MF=L
```

4. Retrieving owner names of all data set profiles

The following example program shows an ICHEINTY coded to retrieve the owner names of all data set profiles in the RACF database.

```
EXAMPLE  CSECT
*
*        entry linkage
*
```

```
         STM   14,12,12(13)              push caller registers
         BALR  12,0                      establish ...
         USING *,12                      ... addressability
         GETMAIN R,LV=DYNLEN             get dynamic storage
         LR    11,1                      move getmained address to R11
         USING DYNAREA,11                addressability to DSECT
         ST    13,SAVEAREA+4             save caller save area address
         LA    15,SAVEAREA               get address of own save area
         ST    15,8(13)                  store in caller save area
         LR    13,15                     get address of own save area
*
*        initialize variables in dynamic storage area
*
         MVC   ENTBLEN,H44               set buffer length to 44
         MVC   ENTNLEN,H1                set entity length to 1
         XC    ENTNAME,ENTNAME           clear entity name area
         MVC   RETALEN,F40               set return area length
*
*        copy static ICHEINTY and ICHEACTN to dynamic GETMAINed areas
*
         MVC   DYNICH(ICHLEN),STATICH
         MVC   DYNACT(ACTLEN),STATACT
         ICHEINTY RELEASE=1.9,ACTIONS=(DYNACT),WKAREA=RETAREA,         *
               OPTIONS=(FLDEF,NOEXEC),GENERIC=NO,MF=(E,DYNICH)
*
*        loop to retrieve all dataset profiles
*          for each high level qualifier, generic profiles are
*            retrieved first
*
LOOP     EQU   *                         start of loop
         XC    RETDATA,RETDATA           clear ICHEINTY return data
         ICHEINTY NEXTC,ENTRYX=ENTBUFF,RELEASE=1.9,MF=(E,DYNICH)
         LTR   15,15                     check return code
         BNZ   DONE                      exit on non zero return code
*               .
*               .
*        process dataset profiles
*               .
*               .
         TM    RETFLAGS,X'80'            check generic bit
         BO    GENERIC                   branch if generic bit is on
         ICHEINTY OPTIONS=(NOEXEC),GENERIC=NO,MF=(E,DYNICH)
         B     LOOP                      process next profile
*
GENERIC  EQU   *                         profile name is generic
         ICHEINTY OPTIONS=(NOEXEC),GENERIC=UNCOND,MF=(E,DYNICH)
         B     LOOP                      process next profile
*
*        return to caller
*
```

```
DONE     EQU   *                         return to caller
         L     13,SAVEAREA+4             caller's save area address
         FREEMAIN R,LV=DYNLEN,A=(11)     get dynamic storage
         LM    14,12,12(13)              pop registers
         SLR   15,15                     clear return code
         BR    14                        return to caller
*
*        static ICHEACTN and ICHEINTY areas
*
STATACT  ICHEACTN FIELD=OWNER
ACTLEN   EQU   *-STATACT                 length of ICHEACTN
*
STATICH  ICHEINTY NEXTC,TYPE='DS',ENTRYX=*-*,RELEASE=1.9,DATAMAP=NEW,  *
               ACTIONS=(STATACT),WKAREA=*-*,MF=L
ICHLEN   EQU   *-STATICH                 length of ICHEINTY
*
*        constants
*
H1       DC    H'1'
H44      DC    H'44'
F40      DC    F'40'
*
*        dynamic area
*
DYNAREA  DSECT
*
SAVEAREA DC    18F'0'
DYNICH   DS    17F                       dynamic ICHEINTY area
DYNACT   DS    6F                        dynamic ICHEACTN area
*
```

```
*        ENTITYX structure
*
ENTBUFF  DS    0CL48
ENTBLEN  DS    H
ENTNLEN  DS    H
ENTNAME  DS    CL44
*
*        return work area
*
RETAREA  DS    0CL40
RETALEN  DS    F                       return area length
RETDATA  DS    0CL36
RETRBA   DS    CL6                     RBA return area
RETFLAGS DS    CL1                     flags
RETRES1  DS    CL1                     reserved
RETDDSC  DS    F                       duplicate dataset name count
RETRES2  DS    CL8                     reserved
RETDLEN  DS    F                       returned data length
RETOWNLN DS    F                       returned owner name length
RETOWNER DS    CL8                     returned owner name
*
DYNLEN   EQU   *-DYNAREA               dynamic area length
*
         END
```

5. Updating the "Installation Fields"

The RACF template defines a repeat group of fields for installation use. There are four of these fields:

USRCNT
Contains the number of repeat members in the group. A repeat member is one USRNM field, one USRDATA field, and one USRFLAG field.

USRNM
Describes the contents of the USRDATA field.

USRDATA
Contains any information that you choose.

USRFLAG
Is a flag associated with USRNM.

The following example shows how the installation fields are used:

```
USRCNT = 2
             USRNM   ACCTNMBR
             USRDATA K83-1234/DQ3
             USRFLG  00

             USRNM   ADDRESS
             USRDATA RFD 4, Box 7711, Phoenicia, NY
             USRFLG  00
```

The following example shows how to add or update a repeat group member. This code will first delete an existing occurrence, based on the name in USRNM, and then add a new occurrence with the desired new (or updated) data. The code is assumed to be preceded by code that initializes the UDATANM, UDATAL1 and UDATAV fields.

In the part of the example not shown, the ACTN3 and ACTN4 macros are addressed by an ICHEINTY-ALTER macro. The ACTN3 and ACTN4 macros must be specified in the ICHEINTY-ACTIONS keyword in the order ACTN3,ACTN4.

```
         ICHEACTN MF=(E,ACTN3),TESTS=TEST3
         ICHETEST MF=(E,TEST3),FLDATA=(,UDATANM)
         ICHEACTN MF=(E,ACTN4),FLDATA=((Rx),UDATA),TESTS=TEST4
         ICHETEST MF=(E,TEST4),FLDATA=(,UDATANM)
            .
            .
            .
         --- Invoke ICHEINTY ---
            .
            .
            .
ACTN3    ICHEACTN FIELD=USRNM,FLDATA='DEL',TESTS=*-*
TEST3    ICHETEST FIELD=USRNM,FLDATA=(8,*-*)                COND=EQ is default.
```

```
ACTN4   ICHEACTN FIELD=USERDATA,FLDATA=(*-*,*-*),TESTS=*-*
TEST4   ICHETEST FIELD=USRNM,FLDATA=(8,*-*),COND=NE
UDATA   DS    0C            Start of USERDATA area.
UDATANM DS    CL8           Contents of USRNM field.
UDATAL1 DS    AL1           Length of USRDATA field.
UDATAV  DS    CL--          Contents of USRDATA field.
*
* The USRFLG field will be at an offset of UDATAL1+1 from
*  the beginning of the UDATAV field.
*
```

Appendix C. ICHNCONV Macro

Note: This information applies to DATASET profiles, which provide protection only on z/OS. RACF for z/VM provides the ability to manage some aspects of DATASET profiles. However, it is only meaningful to manage DATASET profiles when the RACF database is shared with a z/OS system and such database sharing is supported only on z/VM 7.2 and earlier versions. Hence, IBM recommends that you complete the administration and reporting functions from z/OS. If you follow this recommendation, there should be no reason to define a naming convention table on RACF for z/VM.

RACF requires a data set name format where the high-level qualifier of a data set name is a RACF-defined user ID or group name. If your installation's data set naming convention already meets this requirement, you should not have to use this macro.

RACF allows installations to create a naming convention table (ICHNCV00) that RACF uses to check the data set name in all the commands and SVC routines that process data set names. This table helps an installation set up and enforce data set naming conventions that are different from the standard RACF naming conventions.

RACF compares a data set name against each entry in the table until it finds one that matches the name. If RACF does not find a matching entry, the name remains unchanged.

You create a naming convention table, ICHNCV00, by using the ICHNCONV macro. You must assemble the table and link-edit it into RACFLPA LOADLIB.

The table can have up to 400 naming convention entries and can handle data set names of different formats. Each table entry consists of:

- One ICHNCONV DEFINE macro—to start the naming convention and assign it a name
- Zero or more ICHNCONV SELECT macros—to specify the conditions when the naming convention processes the data set name
- Zero or more ICHNCONV ACTION macros—to convert the name to the standard RACF format or to change any of the modifiable variables
- One ICHNCONV END macro—to terminate the naming convention

At the end of all the naming conventions, an ICHNCONV FINAL macro terminates the table itself.

ICHNCONV DEFINE

An ICHNCONV DEFINE macro starts a naming convention and assigns it a name.

The format of the ICHNCONV DEFINE macro is:

```
[label] ICHNCONV  DEFINE,NAME=convention name
```

DEFINE
Identifies the start of a naming convention. The ICHNCONV DEFINE must start each naming convention and there must be only one per convention.

NAME=convention name
Specifies a unique name that you can use for the convention.

The convention name must be 1 to 8 characters long and follow the rules for symbols in assembler language.

ICHNCONV SELECT

An ICHNCONV SELECT macro specifies the conditions when the naming convention processes the data set name.

The format of the ICHNCONV SELECT macro is:

```
[label] ICHNCONV  SELECT,COND=(condition|compound condition)
```

SELECT
Identifies that this convention has selection criteria.

If the condition on the COND parameter is true, the actions on the ICHNCONV ACTION macros will be processed, and processing will continue as specified on the ICHNCONV END macro. If the condition on the COND parameter is not true, RACF bypasses the ICHNCONV ACTION macros and continues with the next convention in the table.

If an ICHNCONV SELECT macro is not coded, RACF unconditionally processes the actions specified on the ICHNCONV ACTION macros, and continues as specified on the ICHNCONV END macro.

All ICHNCONV SELECT macros for a naming convention must follow the ICHNCONV DEFINE macro and precede any ICHNCONV ACTION macros.

COND=(condition)
Specifies the conditions that have to exist before the naming convention processes the data set name.

The "condition" may be a simple comparison condition of the form:

COND=(variable,operator,operand)

You can also use a "compound condition" formed by linking two or more ICHNCONV SELECT macros with logical AND and OR operators:

COND=(variable,operator,operand,AND)

or

COND=(variable,operator,operand,OR)

If a naming convention contains more than one ICHNCONV SELECT macro, all of the SELECT macros except the last must contain either AND or OR to link it to the following macro. The last (or only) ICHNCONV SELECT cannot have AND or OR specified. RACF evaluates compound conditions in the order specified.

variable
Specifies the variables that the convention can reference. Valid variables are:

- GQ — input qualifiers
- G — input qualifier array subscript
- UQ — output qualifiers
- U — output qualifier array subscript
- QUAL — character qualifier
- QCT — initial number of qualifiers
- NAMETYPE — type of data set
- EVENT — event code
- VOLUME — volume serial numbers
- V — volume serial number array subscript
- VCT — number of volumes
- OLDVOL — volume serial of old volume
- WKX — temporary work variable
- WKY — temporary work variable
- WKZ — temporary work variable
- WKA — temporary work variable

- WKB — temporary work variable
- WKC — temporary work variable
- RACUID — caller's user ID
- RACGPID — caller's current connect group

RACF initializes the variables before the first convention. ICHNCONV passes any changes to a variable to subsequent conventions, but only changes made to the variables UQ, QUAL, and NAMETYPE are passed back to the RACF module that called the naming convention table processing module.

You can reference character and hexadecimal variables by substring; for example, (variable,subscript,substring-start,substring-end). If the variable does not accept subscripts or you omit the subscript, you must code a comma to show that the subscript is omitted. Variables cannot be used to define the extents of substrings. For example, (GQ,2,1,3) refers to the first three characters of the second input qualifier; (EVENT,,2,2) refers to the second byte of the event code.

Example: The definition of the data set BOB.SAMPLE.DATASET on volume 111111, when the naming convention table processing module was called during a TSO session when user RACUSR1 was connected to group RACGRP1, would lead to the following set of initial variables:

(GQ,1) = BOB
(GQ,2) = SAMPLE
(GQ,3) = DATASET
(GQ,4) to (GQ,22) = blank
(UQ,0) = blank
(UQ,1) = BOB
(UQ,2) = SAMPLE
(UQ,3) = DATASET
(UQ,4) to (UQ,22) = blank
QCT = 3
QUAL = BOB
NAMETYPE = UNKNOWN
EVENT = X'0201'
(VOLUME,1) = 111111
VCT = 1
G, U, V = -1
WKX, WKY, WKZ = 0
WKA, WKB, WKC = blank
OLDVOL = blank
RACUID = RACUSR1
RACGPID = RACGRP1

GQ
input qualifiers of the data set name.

G
input qualifier array subscript.

UQ
output qualifiers of the data set name.

U
output qualifier array subscript.

GQ and UQ are arrays containing the qualifiers of the data set name with the high-level qualifier of the name in (GQ,1) and (UQ,1). G and U are halfword variables used to hold subscripts to the GQ and UQ arrays. G and U are initialized to negative one (-1), which is out of the range of valid subscripts.

Initially the input and output qualifiers are identical; but if the contents of the output qualifiers are changed, the new contents are used as the new data set name. Each qualifier is an eight-byte character field padded on the right with blanks. (The field does not include the periods that

separate qualifiers.) Initially, (UQ,0) is blank and is reserved for the convention to set as the new high-level qualifier.

If the name produced by the naming conventions table is longer than 44 characters, it is truncated to 44 characters. Thus, the highest possible number of qualifiers (or the highest possible value for the subscripts G and U) is 22.

If you use GQ or UQ in an ICHNCONV SELECT macro without a subscript, RACF tests the condition for each qualifier in turn until the condition is true. The variables G and U are set to the subscript of the qualifier for which the condition was true, G for the conditions using GQ and U for those involving UQ. If the condition is not true, the subscript variable will be negative one (-1).

For all conditions except NE (not equal), the implied linkage is OR; for the NE condition, the implied linkage is AND. For example,

SELECT COND=(GQ,EQ,'ABC') means

SELECT COND=((GQ,0),EQ,'ABC',OR)
SELECT COND=((GQ,1),EQ,'ABC',OR) ...

while

SELECT COND=(GQ,NE,'ABC') means

SELECT COND=((GQ,0),NE,'ABC',AND)
SELECT COND=((GQ,1),NE,'ABC',AND) ...

You may use any numeric variable as a subscript for GQ or UQ. If RACF encounters an out-of-range subscript (for example, -1 or 23), RACF uses blanks for the comparison.

If GQ or UQ is in an ICHNCONV ACTION macro without a subscript, RACF uses the current value of G or U respectively as the subscript.

QUAL
An 8-byte character qualifier that RACF uses in authority checking to determine if the data set is the user's data set or a group data set.

QUAL is initially the data set high-level qualifier. If the high-level qualifier is not a user ID or group name, you should set QUAL to a user ID or group name. Setting QUAL, however, is not the same as setting the data set high-level qualifier. QUAL and the high-level qualifier are two separate fields, used for different RACF processing. Therefore, if you change QUAL, you probably want to set (UQ,0) to the same value as QUAL, especially for generic profile names.

QCT
A 2-byte binary field containing the initial number of qualifiers in the data set name.

NAMETYPE
Indicates whether the data set is a user or group data set.

NAMETYPE initially has the value UNKNOWN but a convention action may set the value to be USER or GROUP. The three special constant values UNKNOWN, USER, and GROUP may be used to test and set the value of this field. NAMETYPE is available only when the caller is RACDEF.

If the convention sets the value to USER or GROUP, RACF ensures that an appropriate user or group exists and fails the RACF or ADDSD if not.

EVENT
A 2-byte hexadecimal field containing the event code that is currently passed to the exit routine.

Values that EVENT may have are:

X'0100' - RACHECK
X'0201' - RACDEF DEFINE (RENAME new name)
X'0202' - RACDEF RENAME (OLD name)
X'0203' - RACDEF ADDVOL

X'0204' - RACDEF DELETE
X'0205' - RACDEF CHGVOL
X'0301' - ADDSD SET
X'0302' - ADDSD NOSET
X'0303' - ADDSD MODEL
X'0401' - ALTDSD SET
X'0402' - ALTDSD NOSET
X'0501' - DELDSD SET
X'0502' - DELDSD NOSET
X'0601' - LISTDSD prelocate (see note)
X'0602' - LISTDSD DATASET
X'0603' - LISTDSD ID or PREFIX
X'0701' - PERMIT TO-resource
X'0702' - PERMIT FROM-resource
X'0801' - SEARCH prelocate (see note)
X'0802' - SEARCH postlocate (see note)
X'0900' - ICHUT100
X'0D00' - RACXTRT

Note: Prelocate means before a profile is located; postlocate means after a profile is located but before it is displayed.

VOLUME
An array of volume serial numbers for volumes containing the data set. Each volume is a 6-byte character field.

V
A 2-byte variable that contains a subscript to the volume array. V is initialized to -1.

VOLUME is not available for generic data set profiles and is not available from commands if the VOLUME keyword was not specified. An attempt to reference nonexistent volumes (subscript 0 or greater than the number of volumes in the VOLUME array) results in a VOLUME parameter which contains *BLANK as a character string.

If you reference VOLUME in an ICHNCONV SELECT macro without a subscript, RACF tests the condition for each volume in turn until the condition is true. The variable V is set to the subscript of the volume for which the condition was true. If the condition is not true, the subscript variable will be negative one (-1).

For all conditions except NE (not equal), the implied linkage is OR; for the NE condition, the implied linkage is AND. For example,

SELECT COND=(VOLUME,EQ,‘ABC’) means

SELECT COND=((VOLUME,1),EQ,‘ABC’,OR)
SELECT COND=((VOLUME,2),EQ,‘ABC’,OR) ...

while

SELECT COND=(VOLUME,NE,‘ABC’) means

SELECT COND=((VOLUME,1),NE,‘ABC’,AND)
SELECT COND=((VOLUME,2),NE,‘ABC’,AND) ...

You may use any numeric variable as a subscript for VOLUME. If VOLUME is in an ICHNCONV ACTION macro without a subscript, RACF uses the current value of V as the subscript.

VCT
A 2-byte binary field containing the number of volumes in the VOLUME array. If volume information is not available, VCT has a value of zero.

OLDVOL
A 6-byte character field containing the volume serial number of the volume that the data set currently resides on. This field is available during a RACDEF ADDVOL or RACDEF CHGVOL request.

WKX, WKY, WKZ
These are 2-byte binary fields that may be used as temporary work variables to save subscripts and other numeric data within and between conventions.

WKA, WKB, WKC
These are 8-byte character fields that may be used as temporary work variables to save qualifiers and other non-numeric data within and between conventions.

RACUID
The caller's user ID.

RACGPID
The caller's current connect group.

operator
Specifies the conditional operator: Valid operators are:

- EQ — Equal
- GT — Greater than
- LT — Less than
- GE — Greater than or equal
- LE — Less than or equal
- NE — Not equal

operand
Specifies a variable, a literal, or one of the following special symbols for use with the NAMETYPE variable:

- USER
- GROUP
- UNKNOWN

The operand used should match the length and type of the variable. If the length does not match, RACF performs padding or truncation in the normal manner. If the type does not match, the results are unpredictable.

If operand is specified as a literal, it can be:

- A character string enclosed in quotes
- A decimal number
- A hexadecimal string in the form X‘string’

ICHNCONV ACTION

An ICHNCONV ACTION macro changes the value of variables. Use these macros to convert the data set name to the standard RACF format. RACF processes the ACTION macros in sequence.

The format of the ICHNCONV ACTION macro is:

```
[label] ICHNCONV  ACTION,SET=(variable,value)
```

ACTION
Identifies a naming convention action. You can code multiple ICHNCONV ACTION macros.

SET=(variable,value)
Changes the qualifiers of a data set name and other variables.

variable
Specifies the variables that the convention can reference and set. See the preceding description of ICHNCONV SELECT for a description of these variables.

The following variables can be set:

- UQ
- QUAL
- G
- U
- V
- NAMETYPE
- WKA, WKB, WKC
- WKX, WKY, WKZ

value
Specifies the value given to the variable. Value can be another variable, a literal, or one of the following special symbols for use with the NAMETYPE variable:

- USER
- GROUP
- UNKNOWN

The value assigned to a variable should match the length and type of the variable. If the length does not match, RACF performs padding or truncation in the normal manner. If the type does not match, the results are unpredictable.

If you specify value as a variable, it can be any of the variables defined in the description of ICHNCONV SELECT.

If value is a literal, it can be:

- A character string enclosed in quotes
- A decimal number
- A hexadecimal string of the form X‘string’

ICHNCONV END

An ICHNCONV END macro terminates the naming convention.

The format of the ICHNCONV END macro is:

```
[label] ICHNCONV  END,NEXT=(convention name|'SUCCESS'|'NEXT'|'ERROR')
```

END
Identifies the end of the naming convention. Each convention must have one ICHNCONV END.

NEXT= (convention name|‘SUCCESS’|‘NEXT’|‘ERROR’)
Specifies where control goes after this convention executes, if the conditions specified in the ICHNCONV SELECT macros have been met or if there are no ICHNCONV SELECT macros in this convention.

If NEXT=convention name, processing continues with the specified convention and skips intervening conventions in the table. The specified convention must not precede the current convention in the table; otherwise, the RACF request fails.

If NEXT=‘NEXT’, processing continues with the next convention in sequence. If NEXT=‘NEXT’ is coded or defaulted to on the last convention in the table, processing is the same as if NEXT=‘SUCCESS’ was coded.

If NEXT='SUCCESS', then the convention processing routine bypasses further convention processing and returns "a successful name processing" return code to the RACF routine that called it. The RACF routine will continue to process normally using the name returned by the convention processing routine.

If NEXT='ERROR', then the convention processing routine bypasses further processing and returns "an invalid data set name" return code to the RACF routine that called it. The RACF routine will terminate processing and fail the request.

ICHNCONV FINAL

An ICHNCONV FINAL macro terminates the naming convention table. (The naming convention table has one ICHNCONV FINAL macro.)

The format of the ICHNCONV FINAL macro is:

```
[label] ICHNCONV  FINAL
```

FINAL
Identifies the end of the naming conventions table. There must be only one ICHNCONV FINAL macro in the table and it must be the last entry in the table.

Example of a Naming Convention Table

The following example of a naming convention table illustrates some ways that a table could be coded.

The first convention checks for data sets that are already in the correct RACF format, with a user ID or group name in the high-level qualifier or system data sets that start with the characters SYS. This convention bypasses all further checks because no further changes are needed.

```
ICHNCONV DEFINE,NAME=CHECK1
ICHNCONV SELECT,COND=((GQ,1),EQ,RACUID,OR)
ICHNCONV SELECT,COND=((GQ,1),EQ,RACGPID,OR)
ICHNCONV SELECT,COND=((GQ,1,1,3),EQ,'SYS')
ICHNCONV END,NEXT='SUCCESS'
```

This convention checks for data set names that have three or more qualifiers and any qualifier is the user's ID. The user ID is moved to the start of the name and deleted from its current position. ICHNCONV sets the type indicator and processing continues with convention CHECK4.

```
ICHNCONV DEFINE,NAME=CHECK2
ICHNCONV SELECT,COND=(QCT,GE,3,AND)
ICHNCONV SELECT,COND=(GQ,EQ,RACUID)
ICHNCONV ACTION,SET=(NAMETYPE,USER)
ICHNCONV ACTION,SET=((UQ,0),(GQ,G)
ICHNCONV ACTION,SET=((UQ,G),'  ')
ICHNCONV END,NEXT=CHECK4
```

For all data sets that did not pass the first two conventions the first four characters of the third and fourth qualifiers are concatenated to form a new fifth qualifier. The user's current connect group becomes a high-level qualifier. Processing continues (by default) with the next convention.

```
ICHNCONV DEFINE,NAME=CHECK3
ICHNCONV ACTION,SET=((UQ,0),RACGRIP)
ICHNCONV ACTION,SET=((UQ,5,1,4),(GQ,3,1,4))
ICHNCONV ACTION,SET=((UQ,5,5,8),(GQ,4,1,4))
ICHNCONV ACTION,SET=(NAMETYPE,GROUP)
ICHNCONV END
```

The installation has decided to enforce a standard that all three-qualifier data set names must have a data set type code as the last qualifier. Any qualifiers that are not in the list will cause the name to be rejected.

```
ICHNCONV DEFINE,NAME=CHECK4
ICHNCONV SELECT,COND=(QCT,EQ,3,AND)
ICHNCONV SELECT,COND=((GQ,3),NE,‘PLI’,AND)
ICHNCONV SELECT,COND=((GQ,3),NE,‘DATA’,AND)
ICHNCONV SELECT,COND=((GQ,3),NE,‘COBOL’,AND)
ICHNCONV SELECT,COND=((GQ,3),NE,‘ASM’)
ICHNCONV END,NEXT=‘ERROR’
```

The ICHNCONV FINAL macro terminates the table. An Assembler END statement is necessary to terminate the assembly.

```
ICHNCONV FINAL
END
```

Appendix D. IBM-Supplied Class Descriptor Table Entries

Programming interface information

- DFTUACC
- GENLIST
- OPER
- POSIT
- RACLIST
- RACLREQ
- RVRSMAC
- SLBLREQ

End programming interface information

Table 188 on page 351 lists the IBM-supplied class entries in the Class Descriptor Table (ICHRRCDX). Other classes can be added to the CDT by your installation.

Table 188. IBM-Supplied Classes

Class		
ACCTNUM	POSIT=126	
	RACLIST=ALLOWED	FIRST=ANY
	GENLIST=DISALLOWED	OTHER=ANY
	OPER=NO	MAXLNTH=39
		DFTUACC=NONE
	ID=46	
ACICSPCT	POSIT=5	
	GROUP=BCICSPCT	FIRST=ANY
		OTHER=ANY
	OPER=NO	MAXLNTH=13
		DFTUACC=NONE
	ID=37	
AIMS	POSIT=4	
		FIRST=ALPHA
		OTHER=ALPHANUM
	OPER=NO	MAXLNTH=8
		DFTUACC=NONE
	ID=11	

Table 188. IBM-Supplied Classes (continued)		
Class		
APPCLU	POSIT=118	
	RACLIST=DISALLOWED	FIRST=ALPHA
	GENLIST=DISALLOWED	OTHER=ANY
	OPER=NO	MAXLNTH=35
		DFTUACC=NONE
	ID=57	
APPCPORT	POSIT=87	PROFDEF=YES
	RACLIST=ALLOWED	FIRST=ALPHA
	GENLIST=DISALLOWED	OTHER=ALPHANUM
	OPER=NO	MAXLNTH=8
		DFTRETC=4
		DFTUACC=NONE
		SLBLREQ=YES
	ID=98	RVRSMAC=YES
APPCSERV	POSIT=84	PROFDEF=YES
	RACLIST=ALLOWED	FIRST=ALPHANUM
	GENLIST=DISALLOWED	OTHER=ANY
	RACLREQ=YES	MAXLNTH=73
		DFTRETC=8
		DFTUACC=NONE
	ID=105	
APPCSI	POSIT=88	PROFDEF=YES
	RACLIST=ALLOWED	FIRST=ALPHANUM
	GENLIST=DISALLOWED	OTHER=ANY
	OPER=NO	MAXLNTH=26
		DFTRETC=4
		DFTUACC=READ
	ID=97	
APPCTP	POSIT=89	PROFDEF=YES
		FIRST=ALPHANUM
	GENLIST=DISALLOWED	OTHER=ANY
	RACLIST=ALLOWED	MAXLENTH=82
		DFTRETC=8
		DFTUACC=NONE
	ID=96	

Table 188. IBM-Supplied Classes (continued)		
Class		
APPL	POSIT=3	
	RACLIST=ALLOWED	FIRST=ALPHA
	GENLIST=ALLOWED	OTHER=ALPHANUM
	OPER=NO	MAXLNTH=8
		DFTUACC=NONE
	ID=8	
BCICSPCT	POSIT=5	
		FIRST=ANY
		OTHER=ANY
	OPER=NO	MAXLNTH=13
		DFTUACC=NONE
	ID=38	
CBIND	POSIT=545	OTHER=ANY
	RACLIST=ALLOWED	MAXLNTH=41
		DFTRETC=8
	OPER=NO	
		ID=1
	FIRST=ALPHA	
CCICSCMD	POSIT=5	
	GROUP=VCICSCMD	FIRST=ANY
		OTHER=ANY
	OPER=NO	MAXLNTH=21
		DFTUACC=NONE
	ID=52	
CIMS	POSIT=93	
	RACLIST=DISALLOWED	FIRST=ALPHA
	GENLIST=DISALLOWED	OTHER=ALPHANUM
	OPER=NO	MAXLNTH=8
		DFTUACC=NONE
	ID=88	

Table 188. IBM-Supplied Classes (continued)

Class		
CONSOLE	POSIT=107	
	RACLIST=ALLOWED	FIRST=ANY
	GENLIST=DISALLOWED	OTHER=ANY
	OPER=NO	MAXLNTH=8
		DFTRETC=8
		DFTUACC=NONE
		RVRSMAC=YES
	ID=68	
CSFKEYS	POSIT=98	
	RACLIST=ALLOWED	FIRST=NONATABC
	GENLIST=DISALLOWED	OTHER=ANY
	RACLREQ=YES	MAXLNTH=73 (1.9.0 or later)
		DFTRETC=4
		DFTUACC=NONE
	ID=100	
CSFSERV	POSIT=98	
	RACLIST=ALLOWED	FIRST=ALPHA
	GENLIST=DISALLOWED	OTHER=ALPHANUM
	RACLREQ=YES	MAXLNTH=8
		DFTRETC=4
		DFTUACC=NONE
	ID=99	
DASDVOL	POSIT=0	
	RACLIST=ALLOWED	FIRST=ALPHANUM
	GENLIST=ALLOWED	OTHER=ALPHANUM
	OPER=YES	MAXLNTH=6
	ID=5	
DCICSDCT	POSIT=5	
	GROUP=ECICSDCT	FIRST=ANY
		OTHER=ANY
	OPER=NO	MAXLNTH=13
		DFTUACC=NONE
	ID=31	

Table 188. IBM-Supplied Classes (continued)

Class		
DEVICES	POSIT=115	
	RACLIST=ALLOWED	FIRST=ANY
	GENLIST=DISALLOWED	OTHER=ANY
	RACLREQ=YES	MAXLNTH=39
		DFTRETC=4
		DFTUACC=NONE
		SLBLREQ=YES
	ID=60	
DIMS	POSIT=93	
	RACLIST=DISALLOWED	FIRST=ALPHA
	GENLIST=DISALLOWED	OTHER=ALPHANUM
	OPER=NO	MAXLNTH=8
		DFTUACC=NONE
	ID=89	
DIRACC	POSIT=71	PROFDEF=NO
	RACLIST=DISALLOWED	FIRST=ANY
	GENLIST=DISALLOWED	OTHER=ANY
	OPER=NO	MAXLNTH=246
		DFTRETC=8
		DFTUACC=NONE
	ID=107	
DIRAUTH	POSIT=105	PROFDEF=NO
	RACLIST=DISALLOWED	FIRST=ANY
	GENLIST=DISALLOWED	OTHER=ANY
	OPER=NO	MAXLNTH=8
		DFTRETC=8
		DFTUACC=NONE
	ID=70	

Table 188. IBM-Supplied Classes (continued)		
Class		
DIRECTRY	POSIT=95	
	RACLIST=DISALLOWED	FIRST=ANY
	GENLIST=DISALLOWED	OTHER=ANY
	OPER=YES	MAXLNTH=153
		DFTRETC=8
		DFTUACC=NONE
		SLBLREQ=YES
	KEYQUAL=2	
	ID=86	
DIRSRCH	POSIT=70	PROFDEF=NO
	RACLIST=DISALLOWED	FIRST=ANY
	GENLIST=DISALLOWED	OTHER=ANY
	OPER=NO	MAXLNTH=246
		DFTRETC=8
		DFTUACC=NONE
	ID=106	
DLFCLASS	POSIT=92	
	RACLIST=ALLOWED	FIRST=ALPHA
	GENLIST=DISALLOWED	OTHER=ANY
	OPER=NO	MAXLNTH=64
		DFTUACC=NONE
	ID=90	
DSNR	POSIT=7	
	RACLIST=ALLOWED	FIRST=ALPHANUM
	GENLIST=ALLOWED	OTHER=ANY
	OPER=NO	MAXLNTH=39
	ID=18	
ECICSDCT	POSIT=5	
		FIRST=ANY
		OTHER=ANY
	OPER=NO	MAXLNTH=13
		DFTUACC=NONE
	ID=32	

Table 188. IBM-Supplied Classes (continued)

Class		
FACILITY	POSIT=8	
	RACLIST=ALLOWED	FIRST=ANY
	GENLIST=ALLOWED	OTHER=ANY
	OPER=NO	MAXLNTH=39
		DFTUACC=NONE
	ID=19	
FCICSFCT	POSIT=5	
	RACLIST=ALLOWED	FIRST=ANY
		OTHER=ANY
	OPER=NO	MAXLNTH=17
		DFTUACC=NONE
	ID=27	
FIELD	POSIT=121	
	RACLIST=ALLOWED	FIRST=ALPHA
	GENLIST=ALLOWED	OTHER=ANY
	OPER=NO	MAXLNTH=26
		DFTUACC=NONE
	ID=51	
FILE	POSIT=94	
	RACLIST=DISALLOWED	FIRST=ANY
	GENLIST=DISALLOWED	OTHER=ANY
	OPER=YES	MAXLNTH=171
		DFTRETC=8
		DFTUACC=NONE
		SLBLREQ=YES
	KEYQUAL=2	
	ID=87	
FIMS	POSIT=101	
	RACLIST=DISALLOWED	FIRST=ALPHANUM
	GENLIST=DISALLOWED	OTHER=ALPHANUM
	OPER=NO	MAXLNTH=8
		DFTUACC=NONE
	ID=79	

Table 188. IBM-Supplied Classes (continued)

Class		
FSOBJ	POSIT=72	PROFDEF=NO
	RACLIST=DISALLOWED	FIRST=ANY
	GENLIST=DISALLOWED	OTHER=ANY
	OPER=NO	MAXLNTH=246
		DFTRETC=8
		DFTUACC=NONE
	ID=108	
FSSEC	POSIT=73	PROFDEF=NO
	RACLIST=DISALLOWED	FIRST=ANY
	GENLIST=DISALLOWED	OTHER=ANY
	OPER=NO	MAXLNTH=246
		DFTRETC=8
		DFTUACC=NONE
	ID=109	
GCICSTRN	POSIT=5	
		FIRST=ANY
		OTHER=ANY
	OPER=NO	MAXLNTH=13
		DFTUACC=NONE
	ID=13	
GCSFKEYS	POSIT=98	
	RACLIST=ALLOWED	FIRST=NONATABC
	GENLIST=DISALLOWED	OTHER=NONATNUM
	RACLREQ=YES	MAXLNTH=17
		DFTRETC=4
		DFTUACC=NONE
	ID=101	
GDASDVOL	POSIT=0	
		FIRST=ALPHANUM
		OTHER=ALPHANUM
	OPER=YES	MAXLNTH=6
	ID=39	

Table 188. IBM-Supplied Classes (continued)		
Class		
GIMS	POSIT=4	
		FIRST=ALPHA
		OTHER=ALPHANUM
	OPER=NO	MAXLNTH=8
		DFTUACC=NONE
	ID=10	
GINFOMAN	POSIT=85	
	RACLIST=ALLOWED	FIRST=ANY
	GENLIST=ALLOWED	OTHER=ANY
	OPER=NO	MAXLNTH=39
	ID=104	
GLOBAL	POSIT=6	
		FIRST=ANY
		OTHER=ANY
	OPER=NO	MAXLNTH=8
		DFTUACC=NONE
	ID=17	
GMBR	POSIT=6	
	GROUP=GLOBAL	FIRST=ANY
		OTHER=ANY
	OPER=NO	MAXLNTH=39
		DFTUACC=NONE
	ID=16	
GSDSF	POSIT=100	
		FIRST=ANY
		OTHER=ANY
	OPER=NO	MAXLNTH=63
		DFTUACC=NONE
	ID=95	

Table 188. IBM-Supplied Classes (continued)

Class		
GTERMINL	POSIT=2	
		FIRST=ALPHANUM
		OTHER=ALPHANUM
	OPER=NO	MAXLNTH=8
	ID=42	
GXFACILI	POSIT=8	OTHER=ANY
	RACLIST=ALLOWED	MAXLNTH=246
	GENLIST=ALLOWED	DFTRETC=4
		DFTUACC=NONE
	MEMBER=XFACILIT	
	OPER=NO	
		ID=1
	FIRST=ALPHANUM	
HCICSFCT	POSIT=5	
		FIRST=ANY
		OTHER=ANY
	OPER=NO	MAXLNTH=17
		DFTUACC=NONE
	ID=28	
HIMS	POSIT=101	
	RACLIST=DISALLOWED	FIRST=ALPHANUM
	GENLIST=DISALLOWED	OTHER=ALPHANUM
	OPER=NO	MAXLNTH=8
		DFTUACC=NONE
	ID=80	
INFOMAN	POSIT=85	
	RACLIST=ALLOWED	FIRST=ANY
	GENLIST=ALLOWED	OTHER=ANY
	OPER=NO	MAXLNTH=39
	ID=103	

Table 188. IBM-Supplied Classes (continued)		
Class		
JCICSJCT	POSIT=5	
	GROUP=KCICSJCT	FIRST=ANY
		OTHER=ANY
	OPER=NO	MAXLNTH=16
		DFTUACC=NONE
	ID=29	
JESINPUT	POSIT=108	
	RACLIST=ALLOWED	FIRST=ANY
	GENLIST=DISALLOWED	OTHER=ANY
	OPER=NO	MAXLNTH=8
		DFTRETC=8
		DFTUACC=NONE
	ID=67	
JESJOBS	POSIT=109	
	RACLIST=ALLOWED	FIRST=ANY
	GENLIST=ALLOWED	OTHER=ANY
	OPER=NO	MAXLNTH=39
		DFTRETC=8
		DFTUACC=NONE
	ID=66	
JESSPOOL	POSIT=110	
	RACLIST=ALLOWED	FIRST=ANY
	GENLIST=DISALLOWED	OTHER=ANY
	OPER=NO	MAXLNTH=53
		DFTRETC=8
		DFTUACC=NONE
	ID=65	
KCICSJCT	POSIT=5	
		FIRST=ANY
		OTHER=ANY
	OPER=NO	MAXLNTH=16
		DFTUACC=NONE
	ID=30	

Table 188. IBM-Supplied Classes (continued)

Class		
MCICSPPT	POSIT=5	
	GROUP=NCICSPPT	FIRST=ANY
		OTHER=ANY
	OPER=NO	MAXLNTH=17
		DFTUACC=NONE
	ID=35	
MGMTCLAS	POSIT=123	
	RACLIST=ALLOWED	FIRST=ALPHA
	GENLIST=DISALLOWED	OTHER=ALPHANUM
	OPER=NO	MAXLNTH=8
		DFTUACC=NONE
	ID=49	
NCICSPPT	POSIT=5	
		FIRST=ANY
		OTHER=ANY
	OPER=NO	MAXLNTH=17
		DFTUACC=NONE
	ID=36	
NODES	POSIT=103	
	RACLIST=ALLOWED	FIRST=ANY
	GENLIST=DISALLOWED	OTHER=ANY
	RACLREQ=YES	MAXLNTH=24
		DFTUACC=NONE
	ID=72	
NODMBR	POSIT=103	
	RACLIST=DISALLOWED	FIRST=ANY
	GENLIST=DISALLOWED	OTHER=ANY
	OPER=NO	MAXLNTH=8
		DFTUACC=NONE
	ID=43	

Table 188. IBM-Supplied Classes (continued)

Class		
NVASAPDT	POSIT=97	
	RACLIST=DISALLOWED	FIRST=ANY
	GENLIST=DISALLOWED	OTHER=ANY
	OPER=NO	MAXLNTH=17
		DFTUACC=NONE
	ID=83	
OIMS	POSIT=101	
	RACLIST=DISALLOWED	FIRST=ALPHANUM
	GENLIST=DISALLOWED	OTHER=ALPHANUM
	OPER=NO	MAXLNTH=8
		DFTUACC=NONE
	ID=81	
OPERCMDS	POSIT=112	
	RACLIST=ALLOWED	FIRST=ANY
	GENLIST=DISALLOWED	OTHER=ANY
	RACLREQ=YES	MAXLNTH=39
		DFTUACC=NONE
	ID=63	
PCICSPSB	POSIT=5	
	GROUP=QCICSPSB	FIRST=ANY
		OTHER=ANY
	OPER=NO	MAXLNTH=17
		DFTUACC=NONE
	ID=14	
PERFGRP	POSIT=125	
	RACLIST=ALLOWED	FIRST=NUMERIC
	GENLIST=DISALLOWED	OTHER=NUMERIC
	OPER=NO	MAXLNTH=3
		DFTUACC=NONE
	ID=47	

Table 188. IBM-Supplied Classes (continued)

Class		
PIMS	POSIT=101	
	RACLIST=DISALLOWED	FIRST=ALPHANUM
	GENLIST=DISALLOWED	OTHER=ALPHANUM
	OPER=NO	MAXLNTH=8
		DFTUACC=NONE
	ID=75	
PMBR	POSIT=13	
	GROUP=PROGRAM	FIRST=ALPHA
		OTHER=ALPHANUM
	OPER=NO	MAXLNTH=39
		DFTUACC=NONE
	ID=40	
PROCESS	POSIT=74	PROFDEF=NO
	RACLIST=DISALLOWED	FIRST=ANY
	GENLIST=DISALLOWED	OTHER=ANY
	OPER=NO	MAXLNTH=246
		DFTRETC=8
		DFTUACC=NONE
	ID=110	
PROGRAM	POSIT=13	
		FIRST=ALPHA
		OTHER=ALPHANUM
	OPER=NO	MAXLNTH=8
		DFTUACC=NONE
	ID=41	
PROPCNTL	POSIT=119	
	RACLIST=ALLOWED	FIRST=ALPHANUM
	GENLIST=DISALLOWED	OTHER=ALPHANUM
	RACLREQ=YES	MAXLNTH=8
		DFTUACC=NONE
	ID=56	

Table 188. IBM-Supplied Classes (continued)		
Class		
PSFMPL	POSIT=113	
	RACLIST=ALLOWED	FIRST=ANY
	GENLIST=DISALLOWED	OTHER=ANY
	RACLREQ=YES	MAXLNTH=39
		DFTRETC=8
		DFTUACC=NONE
	ID=62	
PTKTDATA	POSIT=76	
	RACLIST=ALLOWED	FIRST=ALPHANUM
		OTHER=ANY
	RACLREQ=YES	MAXLNTH=39
		DFTUACC=NONE
	ID=112	
QCICSPSB	POSIT=5	
		FIRST=ANY
		OTHER=ANY
	OPER=NO	MAXLNTH=17
		DFTUACC=NONE
	ID=15	
QIMS	POSIT=101	
	RACLIST=DISALLOWED	FIRST=ALPHANUM
	GENLIST=DISALLOWED	OTHER=ALPHANUM
	OPER=NO	MAXLNTH=8
		DFTUACC=NONE
	ID=76	
RACFEVNT	POSIT=574	
	RACLIST=ALLOWED	FIRST=ANY
	GENLIST=ALLOWED	OTHER=ANY
	RACLREQ=NO	MAXLNTH=246
	OPER=NO	DFTRETC=4
		DFTUACC=NONE
	ID=1	

Table 188. IBM-Supplied Classes (continued)

Class		
RACFVARS	POSIT=102	
	RACLIST=ALLOWED	FIRST=ANY
	GENLIST=DISALLOWED	OTHER=ALPHANUM
	RACLREQ=YES	MAXLNTH=8
		DFTUACC=NONE
	ID=74	
RMTOPS	POSIT=86	
		FIRST=ALPHA
		OTHER=ANY
	OPER=NO	MAXLNTH=246
		DFTUACC=NONE
	ID=102	
RVARSMBR	POSIT=102	
	RACLIST=DISALLOWED	FIRST=ANY
	GENLIST=DISALLOWED	OTHER=ANY
	OPER=NO	MAXLNTH=39
		DFTUACC=NONE
	ID=73	
SCDMBR	POSIT=9	
	GROUP=SECDATA	FIRST=ANY
		OTHER=ANY
	OPER=NO	MAXLNTH=39
		DFTUACC=NONE
	ID=25	
SCICSTST	POSIT=5	
	GROUP=UCICSTST	FIRST=ANY
		OTHER=ANY
	OPER=NO	MAXLNTH=17
		DFTUACC=NONE
	ID=33	

Table 188. IBM-Supplied Classes (continued)		
Class		
SDSF	POSIT=100	
	RACLIST=ALLOWED	FIRST=ANY
	GENLIST=ALLOWED	OTHER=ANY
	OPER=NO	MAXLNTH=63
		DFTUACC=NONE
	ID=94	
SECDATA	POSIT=9	
		FIRST=ALPHA
		OTHER=ALPHA
	OPER=NO	MAXLNTH=8
		DFTUACC=NONE
	ID=26	
SECLABEL	POSIT=117	
	RACLIST=ALLOWED	FIRST=ALPHA
	GENLIST=DISALLOWED	OTHER=ALPHANUM
	RACLREQ=YES	MAXLNTH=8
		DFTRETC=8
		DFTUACC=NONE
	ID=58	
SERVAUTH	POSIT=558	OTHER=ANY
	RACLIST=ALLOWED	MAXLNTH=64
		DFTRETC=4
	RACLREQ=YES	DFTUACC=NONE
		SLBLREQ=YES
		EQUALMAC=YES
	OPER=NO	
		ID=1
	FIRST=ALPHA	
	SIGNAL=YES	

Table 188. IBM-Supplied Classes (continued)

Class		
SFSCMD	POSIT=99	
	RACLIST=DISALLOWED	FIRST=ANY
	GENLIST=DISALLOWED	OTHER=ANY
	OPER=NO	MAXLNTH=246
		DFTRETC=8
		DFTUACC=NONE
	ID=93	
SIMS	POSIT=101	
	RACLIST=DISALLOWED	FIRST=ALPHANUM
	GENLIST=DISALLOWED	OTHER=ALPHANUM
	OPER=NO	MAXLNTH=8
		DFTUACC=NONE
	ID=77	
SMESSAGE	POSIT=116	
	RACLIST=ALLOWED	FIRST=ALPHANUM
	GENLIST=DISALLOWED	OTHER=ALPHANUM
	OPER=NO	MAXLNTH=8
		DFTRETC=0
		DFTUACC=NONE
	ID=59	
STORCLAS	POSIT=122	
	RACLIST=ALLOWED	FIRST=ALPHA
	GENLIST=DISALLOWED	OTHER=ALPHANUM
	OPER=NO	MAXLNTH=8
		DFTUACC=NONE
	ID=50	
SURROGAT	POSIT=104	
	RACLIST=ALLOWED	FIRST=ANY
	GENLIST=DISALLOWED	OTHER=ANY
	OPER=NO	MAXLNTH=17
		DFTUACC=NONE
	ID=71	

Table 188. IBM-Supplied Classes (continued)

Class		
TAPEVOL	POSIT=1	
		FIRST=ALPHANUM
		OTHER=ALPHANUM
	OPER=YES	MAXLNTH=6
		SLBLREQ=YES
	ID=6	
TCICSTRN	POSIT=5	
	GROUP=GCICSTRN	FIRST=ANY
		OTHER=ANY
	OPER=NO	MAXLNTH=13
		DFTUACC=NONE
	ID=12	
TEMPDSN	POSIT=106	PROFDEF=NO
	RACLIST=DISALLOWED	FIRST=ANY
	GENLIST=DISALLOWED	OTHER=ANY
	OPER=NO	MAXLNTH=39
		DFTRETC=8
		DFTUACC=NONE
	ID=69	
TERMINAL	POSIT=2	
	RACLIST=ALLOWED	FIRST=ALPHANUM
	GENLIST=ALLOWED	OTHER=ALPHANUM
	OPER=NO	MAXLNTH=8
		SLBLREQ=YES
		RVRSMAC=YES
	ID=7	
TIMS	POSIT=4	
	GROUP=GIMS	FIRST=ALPHANUM
		OTHER=ALPHANUM
	OPER=NO	MAXLNTH=8
		DFTUACC=NONE
	ID=9	

Table 188. IBM-Supplied Classes (continued)

Class		
TSOAUTH	POSIT=124	
	RACLIST=ALLOWED	FIRST=ALPHANUM
	GENLIST=DISALLOWED	OTHER=ALPHANUM
	OPER=NO	MAXLNTH=8
		DFTUACC=NONE
	ID=48	
TSOPROC	POSIT=127	
	RACLIST=ALLOWED	FIRST=ALPHA
	GENLIST=DISALLOWED	OTHER=ALPHANUM
	OPER=NO	MAXLNTH=8
		DFTUACC=NONE
	ID=45	
UCICSTST	POSIT=5	
		FIRST=ANY
		OTHER=ANY
	OPER=NO	MAXLNTH=17
		DFTUACC=NONE
	ID=34	
UIMS	POSIT=101	
	RACLIST=DISALLOWED	FIRST=ALPHANUM
	GENLIST=DISALLOWED	OTHER=ALPHANUM
	OPER=NO	MAXLNTH=8
		DFTUACC=NONE
	ID=78	
VCICSCMD	POSIT=5	
		FIRST=ANY
		OTHER=ANY
	OPER=NO	MAXLNTH=21
		DFTUACC=NONE
	ID=53	

Table 188. IBM-Supplied Classes (continued)

Class		
VMBATCH	POSIT=15	
	RACLIST=ALLOWED	FIRST=ANY
	GENLIST=ALLOWED	OTHER=ANY
		MAXLNTH=8
		DFTRETC=4
		DFTUACC=NONE
	ID=24	
VMBR	POSIT=120	
	RACLIST=DISALLOWED	FIRST=ALPHA
	GENLIST=DISALLOWED	OTHER=ALPHANUM
	OPER=NO	MAXLNTH=39
		DFTUACC=NONE
	ID=54	
VMCMD	POSIT=14	
	RACLIST=ALLOWED	FIRST=ANY
	GENLIST=ALLOWED	OTHER=ANY
		MAXLNTH=17
		DFTRETC=4
		DFTUACC=NONE
	ID=22	
VMEVENT	POSIT=120	
	RACLIST=DISALLOWED	FIRST=ALPHA
	GENLIST=DISALLOWED	OTHER=ANY
	OPER=NO	MAXLNTH=16
		DFTUACC=NONE
	ID=55	
VMLAN	POSIT=64	OTHER=ANY
	RACLIST=ALLOWED	MAXLNTH=246
	GENLIST=ALLOWED	DFTRETC=4
		DFTUACC=NONE
		SLBLREQ=YES
	OPER=NO	
		ID=1
	FIRST=ANY	

Table 188. IBM-Supplied Classes (continued)		
Class		
VMMAC	POSIT=91	PROFDEF=NO
	RACLIST=DISALLOWED	FIRST=ANY
	GENLIST=DISALLOWED	OTHER=ANY
	OPER=NO	MAXLNTH=246
		DFTRETC=8
		DFTUACC=NONE
		SLBLREQ=YES
	ID=91	
VMMDISK	POSIT=18	
	RACLIST=DISALLOWED	FIRST=ANY
	GENLIST=ALLOWED	OTHER=ANY
		MAXLNTH=22
		DFTUACC=NONE
		SLBLREQ=YES
	ID=20	
VMNODE	POSIT=16	
	RACLIST=ALLOWED	FIRST=ANY
	GENLIST=ALLOWED	OTHER=ANY
		MAXLNTH=8
		DFTRETC=4
		DFTUACC=NONE
	ID=23	
VMPOSIX	POSIT=63	
	RACLIST=DISALLOWED	FIRST=ANY
	GENLIST=DISALLOWED	OTHER=ANY
	OPER=NO	MAXLNTH=246
		DFTRETC=4
		DFTUACC=NONE
	ID=1	
VMRDR	POSIT=17	
	RACLIST=DISALLOWED	FIRST=ANY
	GENLIST=ALLOWED	OTHER=ANY
		MAXLNTH=17
		DFTRETC=4
		DFTUACC=NONE
	ID=21	

Table 188. IBM-Supplied Classes (continued)		
Class		
VMSEGMT	POSIT=90	
	RACLIST=ALLOWED	FIRST=ANY
	GENLIST=ALLOWED	OTHER=ANY
	OPER=NO	MAXLNTH=246
		DFTRETC=4
		DFTUACC=NONE
		SLBLREQ=YES
	ID=92	
VMXEVENT	POSIT=96	
	RACLIST=DISALLOWED	FIRST=ALPHA
	GENLIST=DISALLOWED	OTHER=ANY
	OPER=NO	MAXLNTH=16
		DFTUACC=NONE
	ID=85	
VTAMAPPL	POSIT=114	
	RACLIST=ALLOWED	FIRST=ALPHANUM
	GENLIST=DISALLOWED	OTHER=ALPHANUM
	RACLREQ=YES	MAXLNTH=8
		DFTUACC=NONE
	ID=61	
VXMBR	POSIT=96	
	RACLIST=DISALLOWED	FIRST=ALPHA
	GENLIST=DISALLOWED	OTHER=ALPHANUM
	OPER=NO	MAXLNTH=39
		DFTUACC=NONE
	ID=84	
WIMS	POSIT=101	
	RACLIST=DISALLOWED	FIRST=ALPHANUM
	GENLIST=DISALLOWED	OTHER=ALPHANUM
	OPER=NO	MAXLNTH=8
		DFTUACC=NONE
	ID=82	

Table 188. IBM-Supplied Classes (continued)

Class		
XFACILIT	POSIT=8	OTHER=ANY
	RACLIST=ALLOWED	MAXLNTH=246
	GENLIST=ALLOWED	DFTRETC=8
		DFTUACC=NONE
	GROUP=GXFACILI	
	OPER=NO	
		ID=1
	FIRST=ALPHANUM	
WRITER	POSIT=111	
	RACLIST=ALLOWED	FIRST=ANY
	GENLIST=DISALLOWED	OTHER=ANY
	OPER=NO	MAXLNTH=39
		DFTRETC=8
		DFTUACC=NONE
		SLBLREQ=YES
		RVRSMAC=YES
	ID=64	

Appendix E. List of the Names of Macros Intended for Customer Use

The macros identified in this appendix are provided as programming interfaces for customers by RACF.

Attention: Do not use as programming interfaces any RACF macros other than those identified in this chapter.

General-Use Programming Interfaces

The macros listed in this topic are general-use programming interfaces intended for customer use. Some macros have keywords, fields, or parameters that are designed for IBM internal use only. Such keywords, fields, or parameters are not part of the programming interfaces for use by customers in writing programs that request or receive the services of the security manager. Refer to the appropriate product documentation for the correct classification and use of keywords, fields, and parameters for macros.

Executable Macros

This section lists the executable macros that are general-use programming interfaces for RACF. The standard, execute, list, and modify forms of the macros are programming interfaces.

Table 189. General-Use Executable Macros

Name
ICHEACTN
ICHEINTY
ICHETEST

Mapping Macros

There are no mapping macros that are general-use programming interfaces for RACF.

Product-Sensitive Programming Interface Macros

The macros that are listed in this topic are product-sensitive programming interfaces intended for customer use. Some macros have keywords, fields, or parameters that are designed for IBM internal use only. Such keywords, fields, or parameters are not part of the programming interfaces for use by the customers in writing programs that request or receive the services of the security manager. Refer to the appropriate product documentation for the correct classification and use of keywords, fields, and parameters for macros.

Executable Macros

There are no executable macros that are product-sensitive programming interfaces for RACF.

Mapping Macros

This section lists the mapping macros that are product-sensitive programming interfaces for RACF. The data areas are programming interfaces or contain fields that are programming interfaces. These macros are shipped in the RACF MACLIB file on the RACF service machine's 305 disk.

Table 190. Product-Sensitive Mapping Macros

Macro ID	Name	Macro ID	Name
ICHPWX2	PWX2	ICHRSMXP	RSMXP
ICHCCXP	CCXP	ICHCNXP	CNXP
ICHDEXP	DEXP	ICHPWXP	PWXP
ICHRCXP	RCXP	ICHRDXP	RDXP
ICHRFXP	RFXP	ICHRIXP	RIXP
ICHRLX1P	RLX1P	ICHRLX2P	RLX2P

Appendix F. RACF Database Templates

Note: These templates include fields from z/OS V1R8. Not all of these segments and fields can be created by RACF on z/VM.

Included in this appendix are the following templates:

1. GROUP
2. USER
3. CONNECT
4. DATA SET
5. GENERAL
6. RESERVED

Attention
Do not modify the RACF database templates (CSECT IRRTEMP2). Such modification is not supported by IBM and might result in damage to your RACF database or other unpredictable results.

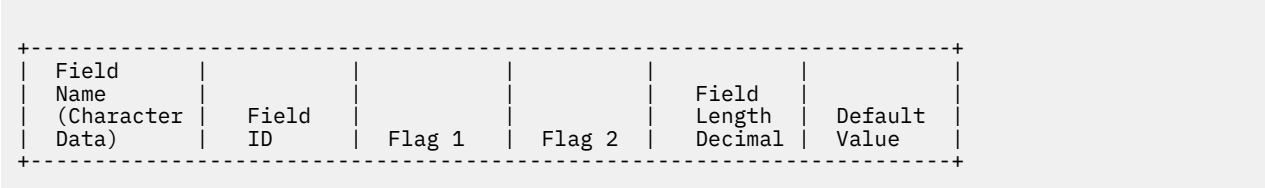
Note: The first field in a segment of a template cannot be retrieved or updated. This field has a Field ID of 001 and is usually described in the **'Field Being Described'** column as 'Start of Segment Fields'.

Format of Field Definitions (RACF Database)

The restructured RACF database templates contain a 31-byte definition for each field in the profile.

Note: The field reference number (Field ID) is now part of the templates.

Each field definition contains information about the field in the following format:



Field Name		**Character data**
Field ID		**Reference Number**
Flag 1 field		**The bits have the following meanings when they are turned on:**
	Bit 0:	The field is a member of a repeat group.
	Bit 1:	The definition describes a combination field.
	Bit 2:	The field is a flag byte.
	Bit 3:	The field contains the count of members in the repeat group following this field.
	Bit 4:	The definition describes a combination field continued in next entry.
	Bit 5:	The field is masked.
	Bit 6:	The field is sorted.
	Bit 7:	The field is a statistical field. A value is always stored for this field, even when it is equal to the defined null value for the field.

Field Name		**Character data**
Field ID		**Reference Number**
Flag 1 field		**The bits have the following meanings when they are turned on:**
Flag 2		The bits have the following meanings when they are turned on:
	Bit 0:	Changes to this field affect security and cause ACEEs to be purged from VLF.
	Bit 1:	The field is padded on the left with binary zeros when values less than the field length are retrieved.
	Bit 2:	This field represents a 3-byte date field.
	Bit 3:	This field is an Application Identity Mapping alias name.
	Bit 4:	This field is not to be unloaded by the Database Unload utility (IRRDBU00).
	Bit 5:	The alias name in this field is EBCDIC.
	Bit 6-7:	Reserved for IBM use.
Field Length		Field length on return from ICHEINTY or RACROUTE REQUEST=EXTRACT (0 is variable length).
Default Value		Field default. If the field is not present in the profile, this byte is propagated throughout the returned field as the default value.
Type		Data type of each field. In this column, character is represented as 'Char', integer is represented as 'Int', and binary is represented as 'Bin'. 'Date' and 'Time' are also possible data types. The type of a combination field that represents a single field is the same as that single field. There is no "type" associated with a combination field which represents multiple fields.

Repeat Groups on the RACF Database

A repeat group consists of one or more sequential fields within a profile that are able to be repeated within that profile. A field that belongs to a repeat group is only defined once in the template, but can be repeated as many times as necessary within the actual profile. A count field precedes the repeat group in the profile indicating how many of these groups follow.

Field Length

If a field in a profile has a fixed length, a value (less than 255) in the field definition within the template specifies its actual length. If a field in a profile has a variable length, the value in the field definition is 0. In both cases, the actual field length is contained in the physical data mapped by the field definition.

Data field types

RACF stores information in the RACF database in many different formats. This section identifies the major data types that RACF stores. Exceptions and additional detail can be found in the description of each specific field within the templates.

Date fields

The format of the 3-byte date fields is *yydddF*, which represents a packed decimal number in which *y* represents year, *d* represents day, and *F* represents the sign. Examples of RACF date values are X'98111C' and X'94099D'.

The format of the 4-byte date fields should be *yyyymmdd*, which represents a packed decimal number in which *y* represents year, *m* represents month, and *d* represents day. Examples of RACF date values are X'19980421' and X'19940409'.

RACF might use any of the following values for null dates: X'FFFFFF', X'00000D', X'00000C', and X'000000' for 3-byte addresses, and X'FFFFFFFF', X'0000000D', X'0000000C', and X'00000000' for 4-byte addresses. However, you should always set null dates to either X'00000F' for 3-byte addresses and X'0000000F' for 4-byte addresses.

Time fields
The format for the 4-byte time fields are *hhmmsstc* where *h* represents hours, *m* represents minutes, *s* represents seconds, *t* represents tenths of seconds, and *c* represents hundredths of seconds. There is no sign byte.

Integer fields
Integers are stored as unsigned binary values. These values can be 1, 2, or 4 bytes in length.

Character fields
Character fields are padded with blanks to the right.

Combination Fields on the RACF Database

The database templates also contain definitions called *combination fields.*

Combination fields do not describe a field of a profile. They contain the field numbers that identify the respective field definitions. You can use the combination field to access multiple fields with one ICHEACTN or RACROUTE REQUEST=EXTRACT macro.

In addition, you can use the combination field to provide aliases for individual fields.

The format of a combination field definition is different from a non-combination definition. Its format is as follows:

Field Name	Character data.
Field ID	Reference number.
Flag 1	The hex representation of the flag bits for this field. For combination fields, bit 1 is on. For a continuation of combination fields, bit 4 is also on.
Flag 2	The hex representation of the flag bits for this field. For combination fields, all bits are off.
Combination IDs	If nonzero, combination IDs represent the position of a non-combination field within the template segment. Up to 5 numbers are allowed.
Comments	Comment field.

Determining Space Requirements for the Profiles

The formula for calculating the space required for each segment (Base RACF information, TSO, DFP, and so on) of each profile in the restructured RACF database is as follows:

```
P = 20 + L + F1 + F4 + R
```

Where:

P	=	The number of bytes required for a profile segment
L	=	The number of bytes in the profile name

F1 = The sum of the lengths of all fields that contain data and have a length of 1 to 127 bytes, plus 2 bytes for every field counted.

For example, if a segment contains 3 non-null fields of length 8, `F1 = (3 * 8) + (3 * 2) = 24 + 6 = 30`.

F4 = The sum of the lengths of all fields that contain data and have a length of 128 to 2**31 bytes, plus 5 bytes for every field counted.

For example, if a segment contains a non-null field 150 bytes long and a non-null field 255 bytes long, `F4 = 150 + 255 + (2 * 5) = 150 + 255 + 10 = 415`

R = The sum of the lengths of all repeat groups. If a repeat group has no occurrences, then it has a length of 0 bytes. If a repeat group has 1 or more occurrences, then the length of each repeat group is calculated as follows:

`9 + N + G1 + G4`

N = The number of occurrences of the group

G1 = The sum of the lengths of all fields in the group, which have a length of 1 to 127 bytes, plus 1 byte for every field counted. If a field has a length of zero, it will still take up 1 byte in the profile.

G4 = The sum of the lengths of all fields in the group, which have a length of 128 to 2**31 bytes, plus 4 bytes for every field counted.

For example, consider a group with two occurrences. Each occurrence contains an 8-byte field and a variable length field. In the first occurrence, the variable length field is 30 bytes and in second occurrence, it is 200 bytes. The length of the group is:

`9 + 2 + G1 + G4`

G1 is `(8 + 1) + (30 + 1)` from the first occurrence and `(8 + 1)` from the second, for a total of 49 bytes. G4 is `(200 + 4)` from the second occurrence, or 204 bytes. So, the length of the group is `9 + 2 + 49 + 204`, or 264 bytes.

Note: For each repeat group the sum of G1 and G4 may not exceed 65535 bytes. For example, this would translate into a maximum of 8191 group connections per user. As another example, this would translate into a maximum of 5957 users connected to a group.

When calculating F1 and F4, remember that statistical fields (Flag1/bit 7 on, in the template definition) are always stored in a profile segment, even when the field contains a null value. For example, REVOKECT

will always add 3 bytes to the length of a USER profile Base segment, regardless of whether it contains a zero value or some other value. Other fields will only exist in the segment, if a specific value has been added for that field.

Note: The RACF database space required for a segment is a multiple of the 256-byte slots required to contain the segment. For example, if a USER profile Base segment contains 188 bytes of data, it will still require 256 bytes of space in the RACF database.

Group template for the RACF database

NOT programming interface information

ACSCNT	FLDNAME
FIELD	FLDVALUE
FLDCNT	INITCNT
FLDFLAG	

End of NOT programming interface information

Note: Application developers should not depend on being able to use RACROUTE REQUEST=EXTRACT for the BASE segment fields on any security product other than RACF. These products are expected to support only such segments as DFP and TSO.

The contents of the group template are as follows:

Template							**Field being described**
Field name (character data)	**Field ID**	**Flag 1**	**Flag 2**	**Field length decimal**	**Default value**	**Type**	
The following is the BASE segment of the GROUP template.							
GROUP	001	00	00	00000000	00		
ENTYPE	002	00	00	00000001	01	Int	The number (1) corresponding to group profiles.
VERSION	003	00	00	00000001	01	Int	The version field from the profile. Always X'01'.
SUPGROUP	004	00	80	00000008	FF	Char	The superior group to this group.
AUTHDATE	005	00	20	00000003	FF	Date	The date the group was created.
AUTHOR	006	00	80	00000008	FF	Char	The owner (user ID or group name) of the group.
INITCNT	007	00	00	00000002	FF		Reserved for IBM's use.
UACC	008	20	00	00000001	00	Bin	The universal group authority. (The authority of a user to the group if the user is not connected to the group.) **Bit** **Meaning when set** **0** JOIN authority **1** CONNECT authority **2** CREATE authority **3** USE authority **4–7** Reserved for IBM's use **Note:** This field has a value of X'00', except for the IBM-defined group VSAMDSET, where the value is X'20'.
NOTRMUAC	009	20	00	00000001	00	Bin	If bit 0 is on, the user must be specifically authorized (by the PERMIT command) to use the terminal. If off, RACF uses the terminal's UACC.
INSTDATA	010	00	00	00000000	00	Char	Installation data.
MODELNAM	011	00	00	00000000	00	Char	Data set model profile name. The profile name begins with the second qualifier; the high-level qualifier is not stored.
FLDCNT	012	10	00	00000004	00		Reserved for IBM's use.

Template							Field being described
Field name (character data)	**Field ID**	**Flag 1**	**Flag 2**	**Field length decimal**	**Default value**	**Type**	
FLDNAME	013	80	00	00000008	00		Reserved for IBM's use.
FLDVALUE	014	80	00	00000000	00		Reserved for IBM's use.
FLDFLAG	015	A0	00	00000001	00		Reserved for IBM's use.
SUBGRPCT	016	10	00	00000004	00	Int	The number of subgroups of the group.
SUBGRPNM	017	80	80	00000008	00	Char	A list of the subgroup names.
ACLCNT	018	10	00	00000004	00	Int	The number of users connected to the group.
USERID	019	80	00	00000008	00	Char	The user ID of each user connected to the group.
USERACS	020	A0	00	00000001	00	Bin	The group authority of each user connected to the group. **Bit** **Meaning when set** **0** JOIN authority **1** CONNECT authority **2** CREATE authority **3** USE authority **4–7** Reserved for IBM's use
ACSCNT	021	80	00	00000002	00		Reserved for IBM's use.
USRCNT	022	10	00	00000004	00	Int	Reserved for installation's use. See **Note 1.**
USRNM	023	80	00	00000008	00		Reserved for installation's use. See **Note 1.**
USRDATA	024	80	00	00000000	00		Reserved for installation's use. See **Note 1.**
USRFLG	025	A0	00	00000001	00		Reserved for installation's use. See **Note 1.**
UNVFLG	026	20	00	00000001	00	Bin	Identifies the group as having (bit 0 is on) or not having the UNIVERSAL attribute.

Note 1: Intended usage for these fields is to allow the installation to store additional data in this profile. USRNM should have a field name to use as a key to identify each unique occurrence of a row in the repeat group. USRDATA and USRFLG hold the data associated with that name. For more information, see "Example 5: Updating the installation fields" in "Examples of ICHEINTY, ICHETEST, and ICHEACTN Macro Usage" in Appendix B of *z/VM: RACF Security Server Macros and Interfaces* .

Field name	**Field ID**	**Flag 1**	**Flag 2**	**Combination field IDs**					**Type**	
The following are the COMBINATION fields.										
DEFDATE	000	40	00	005	000	000	000	000	Char	Alias for AUTHDATE
CREADATE	000	40	00	005	000	000	000	000	Char	Alias for AUTHDATE
OWNER	000	40	00	006	000	000	000	000	Char	Alias for AUTHOR
FIELD	000	40	00	013	014	015	000	000		FLDNAME, FLDVALUE, and FLDFLAG
ACL	000	40	00	019	020	021	000	000		USERID, USERACS, and ACSCNT
USERDATA	000	40	00	023	024	025	000	000		USERNM, USERDATA, and USERFLG

Template							Field Being Described
Field Name (Character Data)	**Field ID**	**Flag 1**	**Flag 2**	**Field Length Decimal**	**Default Value**	**Type**	
The following is the DFP Segment of the GROUP Template.							
DFP	001	00	00	00000000	00		Start of segment
DATAAPPL	002	00	00	00000000	00	Char	Data Application
DATACLAS	003	00	00	00000000	00	Char	Data Class
MGMTCLAS	004	00	00	00000000	00	Char	Management Class
STORCLAS	005	00	00	00000000	00	Char	Storage Class
The following is the OMVS Segment of the GROUP Template.							
OMVS	001	00	00	00000000	00		Start of segment
GID	002	00	10	00000004	FF	Int	GID
The following is the OVM Segment of the GROUP Template.							
OVM	001	00	00	00000000	00		Start of segment
GID	002	00	00	00000004	FF	Int	GID
The following is the TME Segment of the GROUP Template.							
TME	001	00	00	00000000	00		Start of segment fields
ROLEN	002	10	00	00000004	00	Int	Count of roles
ROLES	003	80	00	00000000	00	Char	Role names

User Template for the Restructured Database

The user template describes the fields of the user profiles in a RACF database.

NOT programming interface information

CATEGORY	FLDVALUE	OLDPWDX	PHRCNT
CONGRPCT	MAGSTRIP	OLDPHREX	PPHENV
CONGRPNM	NUMCTGY	OPWDX	PREVKEY
CURKEY	OLDPHR	OPWDXCT	PREVKEYV
CURKEYV	OLDPHRES	OPWDXGEN	PWDCNT
ENCTYPE	OLDPHRNM	PASSWORD	PWDENV
FIELD	OLDPHRX	PHRASE	PWDGEN
FLDCNT	OLDPHRNX	PHRASEX	PWDX
FLDFLAG	OLDPWD	PHRCNTX	SALT
FLDNAME	OLDPWDNM	PHRGEN	

End of NOT programming interface information

Notes:

1. Application developers should not depend on being able to use RACROUTE REQUEST=EXTRACT for the BASE segment fields on any security product other than RACF. These products are expected to support only such segments as DFP and TSO.
2. PASSWORD and PHRASE are not programming interface fields when KDFAES is the active encryption algorithm.

The contents of the user template (base segment) are as follows:

Template							**Field being described**
Field name (character data)	**Field ID**	**Flag 1**	**Flag 2**	**Field length decimal**	**Default value**	**Type**	
The following is the BASE segment of the USER template.							
USER	001	00	00	00000000	00		
ENTYPE	002	00	00	00000001	02	Int	The number (2) corresponding to user profiles.
VERSION	003	00	00	00000001	01	Int	The version field from the profile. Always X'01'.
AUTHDATE	004	00	20	00000003	FF	Date	The date the user was defined to RACF.
AUTHOR	005	00	00	00000008	FF	Char	The owner (user ID or group name) of the user profile.
FLAG1	006	20	80	00000001	00	Bin	Identifies the user as having (bit 0 is on) or not having the ADSP attribute.
FLAG2	007	20	80	00000001	00	Bin	Identifies the user as having (bit 0 is on) or not having the SPECIAL attribute.
FLAG3	008	20	80	00000001	00	Bin	Identifies the user as having (bit 0 is on) or not having the OPERATIONS attribute.
FLAG4	009	20	80	00000001	00	Bin	Identifies the user as having (bit 0 is on) or not having the REVOKE attribute.
FLAG5	010	20	80	00000001	00	Bin	Identifies the user as having (bit 0 is on) or not having the GRPACC attribute.

Template							Field being described
Field name (character data)	**Field ID**	**Flag 1**	**Flag 2**	**Field length decimal**	**Default value**	**Type**	
PASSINT	011	00	80	00000001	FF	Int	The interval in days (represented by a number between 1 and 254) that the user's password is in effect. If it is X'FF', the user's password never expires. See the description of the SETR PASSWORD(INTERVAL...)) processing instructions in *z/VM: RACF Security Server Command Language Reference* for more details.
PASSWORD	012	04	80	00000008	FF	Char	The password associated with the user. For masking, the masked password is stored. For DES or KDFAES, the encrypted user ID is stored. If the installation provides its own password authentication, data returned by the ICHDEX01 exit is stored.
PASSDATE	013	00	20	00000003	FF	Date	The date the password was last changed.
PGMRNAME	014	00	00	00000020	FF	Char	The name of the user.
DFLTGRP	015	00	00	00000008	FF	Char	The default group associated with the user. A value of X'FF' indicates that no group was specified.
LJTIME	016	01	00	00000004	FF	Time	The time that the user last entered the system by using RACROUTE REQUEST=VERIFY.
LJDATE	017	01	20	00000003	FF	Date	The date that the user last entered the system by using RACROUTE REQUEST=VERIFY.
INSTDATA	018	00	80	00000000	00	Char	Installation data.
UAUDIT	019	20	80	00000001	00	Bin	Identifies whether all RACROUTE REQUEST=AUTH, RACROUTE REQUEST=DEFINE, (and, if the caller requests logging, RACROUTE REQUEST=FASTAUTH) macros issued for the user and all RACF commands (except SEARCH, LISTDSD, LISTGRP, LISTUSER, and RLIST) issued by the user will be logged. If bit 0 is on, they are logged. If bit 0 is off, logging might still occur for other reasons, as identified in *z/VM: RACF Security Server Auditor's Guide* .
FLAG6	020	20	80	00000001	00	Bin	Identifies the user as having (bit 0 is on) or not having the AUDITOR attribute.
FLAG7	021	20	80	00000001	00	Bin	If bit 1 is on, this is a protected user ID, which cannot enter the system by any means requiring a password, password phrase, or MFA. If bit 2 is on, this user can enter the system with a password phrase.
FLAG8	022	20	80	00000001	00	Bin	If bit 0 is on, an operator identification card (OID card) is required when logging on to the system. If bit 1 is on, the user must authenticate with MFA unless a LOGON FALLBACK is permitted. If bit 2 is on, the user is permitted to use LOGON FALLBACK to authenticate with other factors such as password or password phrase
MAGSTRIP	023	04	00	00000000	00	Bin	The operator identification associated with the user from the masked or encrypted OID card data required to authenticate this user, as supplied by a supported 327x (such as 3270 and 3278) OID card reader.
PWDGEN	024	00	00	00000001	FF	Int	Current password generation number.
PWDCNT	025	10	00	00000004	00	Int	Number of old passwords present.
OLDPWDNM	026	80	00	00000001	00	Int	Generation number of previous password.

Template							Field being described
Field name (character data)	**Field ID**	**Flag 1**	**Flag 2**	**Field length decimal**	**Default value**	**Type**	
OLDPWD	027	84	00	00000008	FF	Char	Previous password. This is an encrypted password value.
REVOKECT	028	01	80	00000001	FF	Int	Count of unsuccessful password attempts. **Note:** You can use ALTER when setting this field, but you cannot use ALTERI.
MODELNAM	029	00	80	00000000	00	Char	Data set model profile name. The profile name begins with the second qualifier; the high-level qualifier is not stored.
SECLEVEL	030	00	80	00000001	FF	Int	The number that corresponds to the user's security level. For more information on security levels, see *z/OS: Security Server RACF Security Administrator's Guide* .
NUMCTGY	031	10	80	00000004	00	Int	Number of security categories.
CATEGORY	032	80	80	00000002	00	Int	A number that corresponds to the security categories to which the user has access.
REVOKEDT	033	00	20	00000000	00	Date	The date the user will be revoked. This field either has length 0, or contains a 3-byte revoke date.
RESUMEDT	034	00	20	00000000	00	Date	The date the user will be resumed. This field either has length 0, or contains a 3-byte resume date.
LOGDAYS	035	20	00	00000001	00	Bin	The days of the week the user cannot log on (Bit 0 of this field equals Sunday, bit 1 equals Monday, and so on).
LOGTIME	036	00	80	00000000	00	Time	The time of the day the user can log on. If present (length of variable field not equal to 0), it is specified as 6 bytes formatted as two 3-byte packed decimal fields, *0ssssC0eeee*C, where *ssss* represents the start time (*hhmm*) from the ALU...WHEN(TIMES(...)) specification and *eeee* represents the end time. For *hhmm*, *hh* represents hours, and *mm* represents minutes.
FLDCNT	037	10	00	00000004	00		Reserved for IBM's use.
FLDNAME	038	80	00	00000008	00		Reserved for IBM's use.
FLDVALUE	039	80	00	00000000	00		Reserved for IBM's use.
FLDFLAG	040	A0	00	00000001	00		Reserved for IBM's use.
CLCNT	041	10	80	00000004	00	Int	The number of classes in which the user is allowed to define profiles.
CLNAME	042	80	80	00000008	00	Char	A class in which the user is allowed to define profiles. (The user has the CLAUTH attribute.) The user can also define profiles in any other classes with POSIT values matching these classes.
CONGRPCT	043	10	80	00000004	00	Int	The number of groups that the user is connected to.
CONGRPNM	044	80	80	00000008	00	Char	A group that the user is connected to.

Template							Field being described
Field name (character data)	**Field ID**	**Flag 1**	**Flag 2**	**Field length decimal**	**Default value**	**Type**	
USRCNT USRNM USRDATA USRFLG	045 046 047 048	10 80 80 A0	00 80 80 80	00000004 00000008 00000000 00000001	00 00 00 00	Int	Reserved for installation's use. **Note:** Intended usage: For installation to store additional data in this profile. USRNM should have a field name to use as a key to identify each unique occurrence of a row in the repeat group. USRDATA and USRFLG hold the data associated with that name. For more information, see "Example 5: Updating the installation fields" in "Examples of ICHEINTY, ICHETEST, and ICHEACTN Macro Usage" in Appendix B of *z/VM: RACF Security Server Macros and Interfaces* .
SECLABEL	049	00	80	00000008	00	Char	Security label.
CGGRPCT	050	10	80	00000004	00	Int	Number of Connect Group entries. Information from the following CGxxx fields is also available through the logical connect profiles (ICHEINTY with CLASS=CONNECT) in the database. See "Connect Template for the Restructured Database" on page 399 for more details.
CGGRPNM	051	82	80	00000008	00	Char	Connect Group Entry Name.
CGAUTHDA	052	80	A0	00000003	FF	Date	Date the user was connected.
CGAUTHOR	053	80	80	00000008	FF	Char	Owner of connect occurrence.
CGLJTIME	054	81	00	00000004	FF	Time	Time of RACROUTE REQUEST=VERIFY.
CGLJDATE	055	81	20	00000003	FF	Date	Date of RACROUTE REQUEST=VERIFY.
CGUACC	056	A0	80	00000001	00	Bin	Default universal access.
CGINITCT	057	81	00	00000002	FF	Int	Number of RACROUTE REQUEST=VERIFY requests that were successfully processed where the value specified in the CGRPNM field was the current connect group.
CGFLAG1	058	A0	80	00000001	00	Bin	If bit 0 is on, the user has the ADSP attribute in that group.
CGFLAG2	059	A0	80	00000001	00	Bin	If bit 0 is on, the user has the SPECIAL attribute in that group.
CGFLAG3	060	A0	80	00000001	00	Bin	If bit 0 is on, the user has the OPERATIONS attribute in that group.
CGFLAG4	061	A0	80	00000001	00	Bin	If bit 0 is on, the user has the REVOKE attribute in that group.
CGFLAG5	062	A0	80	00000001	00	Bin	If bit 0 is on, the user has the GRPACC attribute in that group.
CGNOTUAC	063	A0	80	00000001	00	Bin	If bit 0 is on, the user must be specifically authorized (by the PERMIT command) to use a terminal. If off, RACF uses the terminal's UACC.
CGGRPAUD	064	A0	80	00000001	00	Bin	If bit 0 is on, the user has the GROUP AUDITOR attribute in that group.
CGREVKDT	065	80	20	00000000	00	Date	The date the user will be revoked. This field either has length 0, or contains a 3-byte revoke date.
CGRESMDT	066	80	20	00000000	00	Date	The date the user will be resumed. This field either has length 0, or contains a 3-byte resume date.
TUCNT	067	10	00	00000002	00	Int	Number of user ID associations.

Template							Field being described
Field name (character data)	**Field ID**	**Flag 1**	**Flag 2**	**Field length decimal**	**Default value**	**Type**	
TUKEY	068	80	00	00000016	00	Char	Associated node and user ID. **Byte** **Meaning when set** **0–7** The associated node name. **8–15** The associated user ID.
TUDATA	069	80	00	00000000			Associated user ID association data **Byte** **Meaning when set** **0** Version number of the TUDATA entry.
						Bin	**1** Bitstring **0** Specifies the user as having (bit is on) or not having (bit is off) a peer user ID association. **1** Specifies the user as being (bit is on) the manager of a managed user ID association. **2** Specifies the user as being (bit is on) managed by a managed user ID association. **3** An association request for this user is pending (bit is on) on a remote RRSF node. **4** An association request for this user is pending (bit is on) on the local RRSF node. **5** Specifies that password synchronization is in effect (bit is on) for this peer-user ID association. **6** Specifies that the association request for this user was rejected (bit is on). **7** Reserved for IBM's use.
							2–20 Reserved for IBM's use.
						Date	**2–24** The date the user ID association was defined. (*yyyymmdd*)
						Time	**25–32** The time the user ID association was defined. For the format of the time, see the TIME macro as documented in *z/OS MVS Programming: Assembler Services Reference, Volume 2 (IARR2V-XCTLX)* .

Template							Field being described
Field name (character data)	**Field ID**	**Flag 1**	**Flag 2**	**Field length decimal**	**Default value**	**Type**	
						Char	**32–36** The date the user ID association was approved or refused. (*yyyymmdd*)
						Int	**37–44** The time the user ID association was approved or refused. For the format of the time, see the TIME macro as documented in *z/OS MVS Programming: Assembler Services Reference, Volume 2 (IARR2V-XCTLX)* .
							45–56 Reserved for IBM's use.
						Char	**57–64** The user ID that created the entry.
CERTCT	070	10	00	00000004	00		Number of certificate names.
CERTNAME	071	80	00	00000000	00		Name of certificate. Names correspond to profiles in the DIGTCERT class for the user.
CERTLABL	072	80	00	00000000	00		Label associated with the certificate.
CERTSJDN	073	80	00	00000000	00		Subject's distinguished name.
CERTPUBK	074	80	00	00000000	00		Public key associated with the certificate.
CERTRSV3	075	80	00	00000000	00		Reserved for IBM's use.
FLAG9	076	20	80	00000001	00		Restricted Access = BIT0.
NMAPCT	077	10	00	00000004	00		Number of DIGTNMAP Mapping Profiles that specify this user ID.
NMAPLABL	078	80	00	00000000	00		Label associated with this mapping.
NMAPNAME	079	80	00	00000000	00		Name of mapping profile. The names correspond to profiles in the DIGTNMAP class.
NMAPRSV1	080	80	00	00000000	00		Reserved for IBM's use.
NMAPRSV2	081	80	00	00000000	00		Reserved for IBM's use.
NMAPRSV3	082	80	00	00000000	00		Reserved for IBM's use.
NMAPRSV4	083	80	00	00000000	00		Reserved for IBM's use.
NMAPRSV5	084	80	00	00000000	00		Reserved for IBM's use.
PWDENV	085	00	08	00000000	00	Bin	Internal form of enveloped RACF password.
PASSASIS	086	20	80	00000001	00	Bin	Identifies the user as having (bit 0 is on) or not having used a mixed case password.
PHRASE	087	04	80	00000000	FF	Bin	The password phrase associated with this user.
PHRDATE	088	00	A0	00000003	FF	Bin	The date the password phrase was last changed.
PHRGEN	089	00	00	00000001	FF	Int	Current password phrase generation number.
PHRCNT	090	10	00	00000004	00	Int	Number of old password phrases.
OLDPHRNM	091	80	00	00000001	00	Int	Generation number of password phrase.
OLDPHR	092	84	00	00000008	FF	Bin	Previous password phrase, truncated to 8 bytes.
CERTSEQN	093	00	00	00000004	00	Int	Sequence number that is incremented whenever a certificate for the user is added, deleted, or altered.
PPHENV	094	00	00	00000000	00	Bin	Internal form of enveloped RACF password phrase

Template							Field being described
Field name (character data)	**Field ID**	**Flag 1**	**Flag 2**	**Field length decimal**	**Default value**	**Type**	
DMAPCT	095	10	00	00000004	00		Number of IDIDMAP Mapping Profiles that specify this user ID.
DMAPLABL	096	80	00	00000000	00		Label associated with this mapping.
DMAPNAME	097	80	00	00000000	00		Name of mapping profile. The names correspond to profiles in the IDIDMAP class.
DMAPRSV1	098	80	00	00000000	00		Reserved for IBM's use.
DMAPRSV2	099	80	00	00000000	00		Reserved for IBM's use.
PWDX	100	04	80	00000000	00	Bin	Password extension
OPWDXCT	101	10	00	00000004	00	Int	Password history extension: count of entries
OPWDXGEN	102	80	00	00000001	FF	Int	Password history extension: generation number
OPWDX	103	84	00	00000000	00	Char	Password history extension: password value
PHRASEX	104	04	00	00000000	00	Bin	Password phrase extension
PHRCNTX	105	10	00	00000004	00	Int	Password phrase history extension: count of entries
OLDPHRNX	106	80	00	00000001	FF	Int	Password phrase history extension: generation number
OLDPHRX	107	84	00	00000000	00	Char	Password phrase history extension: phrase value

Field name	**Field ID**	**Flag 1**	**Flag 2**	**Combination field IDs**					**Type**	
Following are the COMBINATION fields of the USER template										
DEFDATE	000	40	00	004	000	000	000	000		Combination.
CREADATE	000	40	00	004	000	000	000	000		Fields.
OWNER	000	40	00	005	000	000	000	000		
PASSDATA	000	40	00	012	013	000	000	000		
NAME	000	40	00	014	000	000	000	000		
OLDPSWDS	000	40	00	026	027	000	000	000		
LOGINFO	000	40	00	035	036	000	000	000		
FIELD	000	40	00	038	039	040	000	000		
USERDATA	000	40	00	046	047	048	000	000		
CGDEFDAT	000	40	00	052	000	000	000	000		
CGCREADT	000	40	00	052	000	000	000	000		
CGOWNER	000	40	00	053	000	000	000	000		
TUENTRY	000	40	00	068	069	000	000	000		
CERTLIST	000	40	00	071	072	000	000	000		

Field name	Field ID	Flag 1	Flag 2	Combination field IDs					Type	
CERTLST2	000	40	00	071	072	073	074	000		
CERTLST3	000	40	00	071	072	073	000	000		
CERTSIGL	000	40	00	071	073	074	000	000		
OLDPHRES	000	40	00	091	092	000	000	000		
DMAPLST1	000	40	00	096	097	000	000	000		Combination for distributed identity.
OLDPWDX	000	40	00	102	103	000	000	000		Alias for extended password history entry.
OLDPHREX	000	40	00	106	107	000	000	000		Alias for extended password phrase history entry.

Template							Field being described
Field name (character data)	Field ID	Flag 1	Flag 2	Field length decimal	Default value	Type	
Following is the DFP segment of the USER template							
DFP	001	00	00	00000000	00		Start of segment fields
DATAAPPL	002	00	00	00000000	00	Char	Data Application; maximum length=8
DATACLAS	003	00	00	00000000	00	Char	Data Class; maximum length=8
MGMTCLAS	004	00	00	00000000	00	Char	Management Class; maximum length=8
STORCLAS	005	00	00	00000000	00	Char	Storage Class; maximum length=8
Following is the TSO segment of the USER template							
TSO	001	00	00	00000000	00		Start of segment fields
TACCNT	002	00	00	00000000	00	Char	Default account numbers; maximum length=40
TCOMMAND	003	00	00	00000000	00	Char	Default command at logon; maximum length=80
TDEST	004	00	00	00000000	00	Char	Destination identifier; maximum length=8
THCLASS	005	00	00	00000000	00	Char	Default hold class; maximum length=1
TJCLASS	006	00	00	00000000	00	Char	Default job class
TLPROC	007	00	00	00000000	00	Char	Default logon procedure; maximum length=8
TLSIZE	008	00	00	00000004	00	Int	Logon size
TMCLASS	009	00	00	00000000	00	Char	Default message class; maximum length=1
TMSIZE	010	00	00	00000004	00	Int	Maximum region size
TOPTION	011	20	00	00000001	00	Bin	Default for mail notices and OIDcard
TPERFORM	012	00	00	00000004	00	Int	Performance group
TRBA	013	00	00	00000003	00	Bin	RBA of user's broadcast area
TSCLASS	014	00	00	00000000	00	Char	Default sysout class
TUDATA	015	00	00	00000002	00	Bin	2 bytes of hex user data
TUNIT	016	00	00	00000000	00	Char	Default unit name; maximum length=8
TUPT	017	00	00	00000000	00	Bin	Data from UPT control block
TSOSLABL	018	00	00	00000000	00	Char	Default logon SECLABEL; maximum length=8
TCONS	019	00	00	00000000	00	Char	Consoles support
Following is the CICS segment of the USER template							
CICS	001	00	00	00000000	00		Start of segment fields

Template							Field being described
Field name (character data)	**Field ID**	**Flag 1**	**Flag 2**	**Field length decimal**	**Default value**	**Type**	
OPIDENT	002	00	00	00000003	00	Char	Operator identification; 1 to 3 bytes in length
OPCLASSN	003	10	00	00000004	00	Int	Count of operator class values
OPCLASS	004	80	00	00000001	00	Int	Operator class
OPPRTY	005	00	40	00000002	00	Int	Operator priority
XRFSOFF	006	20	00	00000001	00	Bin	XRF Re-signon option: • Bit 0 on = FORCE • Bit 0 off = NOFORCE
TIMEOUT	007	00	40	00000002	00	Bin	Terminal time-out value **Two 1-byte binary fields:** **Note:** 1. first one = hours (0–99) 2. second one = minutes (0–59) 3. special case: hours=0, minutes=60 treated the same as hours=1, minutes=0
RSLKEYN	008	10	00	00000004	00	Int	Count of resource security level (RSL) key values
RSLKEY	009	80	00	00000002	00	Int	RSL key value
TSLKEYN	010	10	00	00000004	00	Int	Count of transaction security level (TSL) key values
TSLKEY	011	80	00	00000002	00	Int	TSL key value
Following is the LANGUAGE segment of the USER template							
LANGUAGE	001	00	00	00000000	00		Start of segment fields
USERNL1	002	00	80	00000003	00	Char	User's primary language
USERNL2	003	00	80	00000003	00	Char	User's secondary language
Following is the OPERPARM segment of the USER template							
OPERPARM	001	00	00	00000000	00		Start of segment fields
OPERSTOR	002	00	00	00000002	00	Bin	STORAGE keyword
OPERAUTH	003	00	00	00000002	00	Bin	AUTH keyword: • X'8000' = MASTER • X'4000' = ALL • X'2000' = SYS • X'10000' = IO • X'0800' = CONS • X'0400' = INFO
OPERMFRM	004	00	00	00000002	00	Bin	MFORM keyword: • Bit 0 indicates T • Bit 1 indicates S • Bit 2 indicates J • Bit 3 indicates M • Bit 4 indicates X

Template							Field being described
Field name (character data)	**Field ID**	**Flag 1**	**Flag 2**	**Field length decimal**	**Default value**	**Type**	
OPERLEVL	005	00	00	00000002	00	Bin	LEVEL keyword: • Bit 0 indicates R • Bit 1 indicates I • Bit 2 indicates CE • Bit 3 indicates E • Bit 4 indicates IN • Bit 5 indicates NB • Bit 6 indicates ALL Bit 6 is mutually exclusive with all other bits except Bit 5.
OPERMON	006	00	00	00000002	00	Bin	MONITOR keyword: • Bit 0 indicates JOBNAMES • Bit 1 indicates JOBNAMEST • Bit 2 indicates SESS • Bit 3 indicates SESST • Bit 4 indicates STATUS Bits 0 and 1 are mutually-exclusive, as are bits 2 and 3.
OPERROUT	007	00	00	00000000	00	Bin	ROUTCODE keyword; 16-bit length bitstring in which each bit indicates a particular ROUTCODE.
OPERLOGC	008	00	00	00000001	00	Bin	LOGCMDRESP keyword. **Value** **Meaning when set** **X'80'** Indicates SYSTEM was specified. **X'40'** Indicates NO was specified.
OPERMGID	009	00	00	00000001	00	Bin	MIGID keyword. **Value** **Meaning when set** **X'80'** Indicates YES was specified. **X'40'** Indicates NO was specified.
OPERDOM	010	00	00	00000001	00	Bin	DOM keyword. **Value** **Meaning when set** **X'80'** Indicates NORMAL was specified. **X'40'** Indicates ALL was specified. **X'20'** Indicates NONE was specified.
OPERKEY	011	00	00	00000000	00	Bin	KEY keyword; maximum length=8
OPERCMDS	012	00	00	00000000	00	Bin	CMDSYS keyword; maximum length=8 (or '*')

Template							Field being described
Field name (character data)	**Field ID**	**Flag 1**	**Flag 2**	**Field length decimal**	**Default value**	**Type**	
OPERUD	013	00	00	00000001	00	Bin	UD keyword. **Value** **Meaning when set** **X'80'** Indicates YES was specified. **X'40'** Indicates NO was specified.
OPERMCNT	014	10	00	00000004	00	Bin	Count of MSCOPE systems
OPERMSCP	015	80	00	00000008	00	Bin	MSCOPE systems
OPERALTG	016	00	00	00000000	00	Bin	ALTGRP keyword **Value** **Meaning when set** **X'80'** Indicates YES was specified. **X'40'** Indicates NO was specified.
OPERAUTO	017	00	00	00000001	00	Bin	AUTO keyword; X'80' indicates YES; X'40' indicates NO.
OPERHC	018	00	00	00000001	00	BIN	HC Keyword; X'80' indicates YES; X'40' indicates NO.
OPERINT	019	00	00	00000001	00	BIN	INTIDS Keyword; X'80' indicates YES; X'40' indicates NO.
OPERUNKN	020	00	00	00000001	00	BIN	UNKNIDS Keyword; X'80' indicates YES; X'40' indicates NO.
Following is the WORK ATTRIBUTES segment of the USER template							
WORKATTR	001	00	80	00000000	00		Start of segment fields
WANAME	002	00	80	00000000	00	Char	User name for SYSOUT; maximum length=60
WABLDG	003	00	80	00000000	00	Char	Building for delivery; maximum length=60
WADEPT	004	00	80	00000000	00	Char	Department for delivery; maximum length=60
WAROOM	005	00	80	00000000	00	Char	Room for delivery; maximum length=60
WAADDR1	006	00	80	00000000	00	Char	SYSOUT address line 1; maximum length=60
WAADDR2	007	00	80	00000000	00	Char	SYSOUT address line 2; maximum length=60
WAADDR3	008	00	80	00000000	00	Char	SYSOUT address line 3; maximum length=60
WAADDR4	009	00	80	00000000	00	Char	SYSOUT address line 4; maximum length=60
WAACCNT	010	00	80	00000000	00	Char	Account number; maximum length=255
Following is the OMVS segment of the USER template							
OMVS	001	00	00	00000000	00		Start of segment fields
UID	002	00	10	00000004	FF	Int	UID
HOME	004	00	00	00000000	00	Char	HOME Path; maximum length=1023
PROGRAM	005	00	00	00000000	00	Char	Initial Program; maximum length=1023
CPUTIME	006	00	00	00000004	FF	Int	CPUTIMEMAX
ASSIZE	007	00	00	00000004	FF	Int	ASSIZEMAX
FILEPROC	008	00	00	00000004	FF	Int	FILEPROCMAX
PROCUSER	009	00	00	00000004	FF	Int	PROCUSERMAX
THREADS	010	00	00	00000004	FF	Int	THREADSMAX

Template							Field being described
Field name (character data)	**Field ID**	**Flag 1**	**Flag 2**	**Field length decimal**	**Default value**	**Type**	
MMAPAREA	011	00	00	00000004	FF	Int	MMAPAREAMAX
MEMLIMIT	012	00	00	00000000	0	Char	MEMLIMIT; maximum length = 9
SHMEMMAX	013	00	00	00000000	0	Char	SHMEMMAX; maximum length = 9
Following is the NETVIEW segment of the USER template							
NETVIEW	001	00	00	00000000	00		Start of segment fields
IC	002	00	00	00000000	00	Char	The command or command list to be processed by NetView® for this operator when the operator logs on to Netview; maximum length=255.
CONSNAME	003	00	00	00000000	00	Char	The default MCS console identifier; maximum length=8.
CTL	004	20	00	00000001	00	Bin	CTL keyword – Specifies whether a security check is performed for this NetView operator when they try to use a span or try to do a cross-domain logon. **Value** **Meaning when set** **X'00'** Indicates CTL was not specified or CTL(SPECIFIC) was specified. **X'80'** Indicates CTL(GLOBAL) was specified. **X'40'** Indicates CTL(GENERAL) was specified.
MSGRECVR	005	20	00	00000001	00	Bin	MSGRECVR keyword **Value** **Meaning when set** **X'00'** Indicates the operator can receive unsolicited messages that are not routed to a specific NetView operator. **X'80'** Indicates the operator cannot receive unsolicited messages that are not routed to a specific NetView operator.
OPCLASSN	006	10	00	00000004	00	Int	Count of operator class values.
OPCLASS	007	80	40	00000002	00	Int	Specifies a NetView scope class for which the operator has authority. This is a 2-byte repeating field. Each member can have fixed-binary values from 1 to 2040.
DOMAINSN	008	10	00	00000004	00	Int	The number of domains the NetView operator controls.
DOMAINS	009	80	00	00000000	00	Char	Specifies the identifier of NetView programs in another NetView domain for which this operator has authority. This is a variable length (5-character maximum) repeating field.

Template							Field being described
Field name (character data)	**Field ID**	**Flag 1**	**Flag 2**	**Field length decimal**	**Default value**	**Type**	
NGMFADMN	010	20	00	00000001	00	Bin	NGMFADMN keyword **Value** **Meaning when set** **X'00'** The NetView operator does not have administrator authority to the NetView Graphic Monitor Facility (NGMF). **X'80'** The NetView operator has administrator authority to the NetView graphic monitor facility (NGMF).
NGMFVSPN	011	00	00	00000000	00		NetView Graphic Monitor Facility view span options; maximum length=8
Following is the DCE segment of the USER template							
DCE	001	00	00	00000000	00		Start of segment fields
UUID	002	00	00	00000036	FF	Char	User's DCE principal's UUID; exactly 36 characters, in the format *nnnnnnnn-nnnn-nnnn-nnnn-nnnnnnnnnnnn* where *n* is any hexadecimal digit.
DCENAME	003	00	00	00000000	00	Char	User's DCE principal name; maximum length=1023
HOMECELL	004	00	00	00000000	00	Char	Home cell for this DCE user; maximum length=1023, and it must start with either */.../* or */.:/*
HOMEUUID	005	00	00	00000036	FF	Char	Home cell UUID; exactly 36 characters, in the format *nnnnnnnn-nnnn-nnnn-nnnn-nnnnnnnnnnnn* where *n* is any hexadecimal digit.
DCEFLAGS	006	20	00	00000001	00	Bin	User flags
DPASSWDS	007	00	00	00000000	00	Char	Current DCE password
DCEENCRY	008	00	00	00000071	00	Bin	PW mask/encrypt key
Following is the OVM segment of the USER template							
OVM	001	00	00	00000000	00		Start of segment fields
UID	002	00	00	00000004	FF	Int	OVM - UID
HOME	003	00	00	00000000	00	Char	Home path; maximum length=1023
PROGRAM	004	00	00	00000000	00	Char	Initial program; maximum length=1023
FSROOT	005	00	00	00000000	00	Char	File system root; maximum length=1023
Following is the LNOTES segment of the USER template							
LNOTES	001	00	00	00000000	00		Start of segment fields
SNAME	002	00	14	00000000	00	Char	User's short name; maximum length=64
Following is the NDS segment of the USER template							
NDS	001	00	00	00000000	00		Start of segment fields
UNAME	002	00	14	00000000	00	Char	User's user name; maximum length=246
Following is the KERB segment of the USER template							
KERB	001	00	00	00000000	00		Start of segment fields
KERBNAME	002	00	00	00000000	00	Char	Kerberos principal name
MINTKTLF	003	00	00	00000000	00	Char	Reserved for IBM's use.
MAXTKTLF	004	00	00	00000000	00	Char	Maximum ticket life

Template							Field being described
Field name (character data)	**Field ID**	**Flag 1**	**Flag 2**	**Field length decimal**	**Default value**	**Type**	
DEFTKTLF	005	00	00	00000000	00	Char	Reserved for IBM's use.
SALT	006	00	00	00000000	00	Char	Current key salt
ENCTYPE	007	00	00	00000000	00	Char	Encryption type
CURKEYV	008	00	00	00000000	00	Char	Current key version
CURKEY	009	00	00	00000000	00	Char	Current DES key
PREVKEYV	010	00	00	00000000	00	Char	Previous key version
PREVKEY	011	00	00	00000000	00	Char	Previous DES key
ENCRYPT	012	00	00	00000004	55	Bin	Encryption types for SETROPTS KERBLVL(1)
Following is the PROXY segment of the USER template							
PROXY	001	00	00	00000000	00		Start of segment fields
LDAPHOST	002	00	00	00000000	00	Char	LDAP server URL; maximum length: 1023
BINDDN	003	00	00	00000000	00	Char	Bind distinguished name; maximum length: 1023
BINDPW	004	00	08	00000000	00	Char	Bind password; maximum length: 128
BINDPWKY	005	00	08	00000071	00	Char	Bind password mask or encrypt key
Following is the EIM segment of the USER template							
EIM	001	00	00	00000000	00	Char	Start of segment fields
LDAPPROF	1	00	00	00000000	00	Char	LDAPBIND profile name

Connect Template for the Restructured Database

The connect template is included to maintain compatibility with previous releases. You can continue to code macros to manipulate CONNECT data. This template is provided to show what fields continue to be supported. Information that was formerly stored in CONNECT profiles was moved to the USER profile. The information is in the CGGRPCT repeat group, and the fields are prefixed by “CG”.

NOT programming interface information

REVOKEDT
RESUMEDT

End of NOT programming interface information

Note: Application developers should not depend on being able to use RACROUTE REQUEST=EXTRACT for the BASE segment fields on any security product other than RACF. These products are expected to support only such segments as DFP and TSO.

The contents of the connect template are as follows:

Template							**Field being described**
Field name (character data)	**Field ID**	**Flag 1**	**Flag 2**	**Field length decimal**	**Default value**	**Type**	
CONNECT	001	00	00	00000000			
ENTYPE	002	00	00	00000001		Int	The number (3) corresponding to connect profiles.
VERSION	003	00	00	00000001		Int	The version field from the profile. Always X'01'.
AUTHDATE	004	00	A0	00000003		Date	The date the user was connected to the group.
AUTHOR	005	00	80	00000008		Char	The owner (user ID) of the connect entry.
LJTIME	006	01	00	00000004		Time	The time that RACROUTE REQUEST=VERIFY was last issued for this user and group.
LJDATE	007	01	20	00000003		Date	The date that RACROUTE REQUEST=VERIFY was last issued for this user and group.
UACC	008	20	80	00000001		Bin	The default universal access authority assigned to the user for this group. **Bit** — **Meaning when set** **0** — ALTER access **1** — CONTROL access **2** — UPDATE access **3** — READ access **4** — EXECUTE access **5–6** — Reserved for IBM's use **7** — EXECUTE access
INITCNT	009	01	00	00000002		Int	The number of RACROUTE REQUEST=VERIFY macro instructions issued for this user and group.
FLAG1	010	20	80	00000001		Bin	Identifies the user as having (bit 0 is on) or not having the ADSP attribute.

Template							Field being described
Field name (character data)	**Field ID**	**Flag 1**	**Flag 2**	**Field length decimal**	**Default value**	**Type**	
FLAG2	011	20	80	00000001		Bin	Identifies the user as having (bit 0 is on) or not having the group-SPECIAL attribute.
FLAG3	012	20	80	00000001		Bin	Identifies the user as having (bit 0 is on) or not having the group-OPERATIONS attribute.
FLAG4	013	20	80	00000001		Bin	Identifies the user as having (bit 0 is on) or not having the REVOKE attribute.
FLAG5	014	20	80	00000001		Bin	Identifies the user as having (bit 0 is on) or not having the GRPACC attribute.
NOTRMUAC	015	20	80	00000001		Bin	Identifies whether the user must be authorized by the PERMIT command with at least READ authority to access a terminal. (If not, RACF uses the terminal's universal access authority.) If bit 0 is on, the user must be specifically authorized to use the terminal.
GRPAUDIT	016	20	80	00000001		Bin	Identifies the user as having (bit 0 is on) or not having the group-AUDITOR attribute.
REVOKEDT	017	00	20	00000000		Date	The date the user will be revoked. This field either has length 0, or contains a 3-byte revoke date.
RESUMEDT	018	00	20	00000000		Date	The date the user will be resumed. This field either has length 0, or contains a 3-byte resume date.

Field name	**Field ID**	**Flag 1**	**Flag 2**	**Combination field IDs**					**Type**	
The following are the COMBINATION fields.										
DEFDATE	000	40	00	004	000	000	000	000	Char	Combination.
CREADATE	000	40	00	004	000	000	000	000	Char	Fields.
OWNER	000	40	00	005	000	000	000	000	Char	

Data Set Template for the Restructured Database

The data set template describes the fields of the data set profiles in a RACF database.

NOT programming interface information

ACL2VAR
AUDITQF
AUDITQS
CATEGORY
FIELD
FLDCNT
FLDFLAG
FLDNAME
FLDVALUE
NUMCTGY

End of NOT programming interface information

Note: Application developers should not depend on being able to use RACROUTE REQUEST=EXTRACT for the BASE segment fields on any security product other than RACF. These products are expected to support only such segments as DFP and TSO.

The contents of the data set template are as follows:

Template							**Field being described**
Field name (character data)	**Field ID**	**Flag 1**	**Flag 2**	**Field length decimal**	**Default value**	**Type**	
DATASET	001	00	00	00000000	00		
ENTYPE	002	00	00	00000001	04	Int	The number (4) corresponding to data set profiles.
VERSION	003	00	00	00000001	01	Int	The version field from the profile. Always X'01'.
CREADATE	004	00	20	00000003	FF	Date	The date the data set was initially defined to RACF; 3-byte date.
AUTHOR	005	00	00	00000008	FF	Char	The owner of the data set.
LREFDAT	006	01	20	00000003	FF	Date	The date the data set was last referenced; 3-byte date.
LCHGDAT	007	01	20	00000003	FF	Date	The date the data set was last updated; 3-byte date.
ACSALTR	008	01	00	00000002	FF	Int	The number of times the data set was accessed with ALTER authority.
ACSCNTL	009	01	00	00000002	FF	Int	The number of times the data set was accessed with CONTROL authority.
ACSUPDT	010	01	00	00000002	FF	Int	The number of times the data set was accessed with UPDATE authority.
ACSREAD	011	01	00	00000002	FF	Int	The number of times the data set was accessed with READ authority.

Template							Field being described
Field name (character data)	**Field ID**	**Flag 1**	**Flag 2**	**Field length decimal**	**Default value**	**Type**	
UNIVACS	012	20	00	00000001	00	Bin	The universal access authority for the data set. **Bit** **Meaning when set** **0** ALTER access **1** CONTROL access **2** UPDATE access **3** READ access **4** EXECUTE access **5–6** Reserved for IBM's use **7** EXECUTE access
FLAG1	013	20	00	00000001	00	Bin	Identifies whether the data set is a group data set. If bit 0 is on, the data set is a group data set.
AUDIT	014	20	00	00000001	00	Bin	Audit Flags. **Bit** **Meaning when set** **0** Audit all accesses **1** Audit successful accesses **2** Audit accesses that fail **3** No auditing **4–7** Reserved for IBM's use
GROUPNM	015	00	00	00000008	FF	Char	The current connect group of the user who created this data set.
DSTYPE	016	20	00	00000001	00	Bin	Identifies the data set as a VSAM, non-VSAM (or generic), MODEL or TAPE data set. **Bit** **Meaning when set** **0** VSAM data set (non-VSAM if this bit is set to 0) **1** MODEL profile **2** Type = TAPE when set on **3–7** Reserved for IBM's use
LEVEL	017	00	00	00000001	FF	Int	Data set level.
DEVTYP	018	00	00	00000004	FF	Bin	The type of device on which the data set resides; only for non-model, discrete data sets.
DEVTYPX	019	00	00	00000008	FF	Char	The EBCDIC name of the device type on which the data set resides; only for non-model, discrete data sets.

Template							Field being described
Field name (character data)	**Field ID**	**Flag 1**	**Flag 2**	**Field length decimal**	**Default value**	**Type**	
GAUDIT	020	20	00	00000001	00	Bin	Global audit flags. (Audit options specified by a user with the AUDITOR or group-AUDITOR attribute.) **Bit** **Meaning when set** **0** Audit all accesses **1** Audit successful accesses **2** Audit accesses that fail **3** No auditing **4–7** Reserved for IBM's use
INSTDATA	021	00	00	00000000	00	Char	Installation data; maximum length=255.
GAUDITQF	025	00	00	00000001	FF	Bin	Global audit FAILURES qualifier. The AUDITQS, AUDITQF, GAUDITQS, and GAUDITQF fields have the following format: **Value** **Meaning when set** **X'00'** Log access at READ level **X'01'** Log access at UPDATE level **X'02'** Log access at CONTROL level **X'03'** Log access at ALTER level
AUDITQS	022	00	00	00000001	FF	Bin	Audit SUCCESS qualifier.
AUDITQF	023	00	00	00000001	FF	Bin	Audit FAILURES qualifier.
GAUDITQS	024	00	00	00000001	FF	Bin	Global audit SUCCESS qualifier.
WARNING	026	20	00	00000001	00	Bin	Identifies the data set as having (bit 7 is on) or not having the WARNING attribute.
SECLEVEL	027	00	00	00000001	FF	Int	Data set security level.
NUMCTGY	028	10	00	00000004	00	Int	The number of categories.
CATEGORY	029	80	00	00000002	00	Bin	A list of numbers corresponding to the categories to which this data set belongs.
NOTIFY	030	00	00	00000000	00	Char	User to be notified when access violations occur against a data set protected by this profile.
RETPD	031	00	00	00000000	00	Int	The number of days protection is provided for the data set. If used, the field will be a two-byte binary number.
ACL2CNT	032	10	00	00000004	00	Int	The number of program and user combinations currently authorized to access the data set.
PROGRAM	033	80	00	00000008	00	Char	The name of a program currently authorized to access the data set, or a 1-byte flag followed by 7 bytes reserved for IBM's use.
USER2ACS	034	80	00	00000008	00	Char	User ID or group.
PROGACS	035	80	00	00000001	00	Bin	The access authority of the program and user combinations.

Template							Field being described
Field name (character data)	**Field ID**	**Flag 1**	**Flag 2**	**Field length decimal**	**Default value**	**Type**	
PACSCNT	036	80	00	00000002	00	Int	Access count.
ACL2VAR	037	80	00	00000000	00	Char	Additional conditional data, 9-byte length, in which the the 1st byte tells what kind of access is allowed and the remaining 8 bytes contain the data.
FLDCNT	038	10	00	00000004	00		Reserved for IBM's use.
FLDNAME	039	80	00	00000008	00		Reserved for IBM's use.
FLDVALUE	040	80	00	00000000	00		Reserved for IBM's use.
FLDFLAG	041	A0	00	00000001	00		Reserved for IBM's use.
VOLCNT	042	10	00	00000004	00	Int	The number of volumes containing the data set.
VOLSER	043	80	00	00000006	00	Char	A list of the serial numbers of the volumes containing the data set.
ACLCNT	044	10	00	00000004	00	Int	The number of users and groups currently authorized to access the data set.
USERID	045	80	00	00000008	00	Char	The user ID or group name of each user or group authorized to access the data set.
USERACS	046	A0	00	00000001	00	Bin	The access authority that each user or group has for the data set.
							Bit **Meaning when set** **0** ALTER access **1** CONTROL access **2** UPDATE access **3** READ access **4** EXECUTE access **5–6** Reserved for IBM's use **7** NONE access
ACSCNT	047	80	00	00000002	00	Int	The number of times the data set was accessed by each user or group.
USRCNT	048	10	00	00000004	00	Int	Reserved for installation's use.
USRNM	049	80	00	00000008	00		Reserved for installation's use.
USRDATA	050	80	00	00000000	00		Reserved for installation's use.
USRFLG	051	A0	00	00000001	00		Reserved for installation's use.
SECLABEL	052	00	00	00000008	00	Char	Security label.

Field name	**Field ID**	**Flag 1**	**Flag 2**	**Combination field IDs**					**Type**	
Following are the COMBINATION fields of the Data Set Template										
DEFDATE	000	40	00	004	000	000	000	000	Char	Combination.
AUTHDATE	000	40	00	004	000	000	000	000	Char	Fields.
OWNER	000	40	00	005	000	000	000	000	Char	
UACC	000	40	00	012	000	000	000	000		

Field name	Field ID	Flag 1	Flag 2	Combination field IDs					Type
ACL2	000	40	00	033	034	035	036	037	
ACL2A3	000	40	00	033	034	035	037	000	
ACL2A2	000	40	00	033	034	035	036	000	
ACL2A1	000	40	00	033	034	035	000	000	
FIELD	000	40	00	039	040	041	000	000	
VOLUME	000	40	00	043	000	000	000	000	
ACL	000	40	00	045	046	047	000	000	
ACL1	000	40	00	045	046	000	000	000	
USERDATA	000	40	00	049	050	051	000	000	

Template							Field being described
Field name (character data)	**Field ID**	**Flag 1**	**Flag 2**	**Field length decimal**	**Default value**	**Type**	
Following is the DFP segment of the Data Set Template							
DFP	001	00	00	00000000	00		Start of segment fields
RESOWNER	002	00	00	00000008	FF	Char	Resource owner; must represent a user ID or group name

Template							Field being described
Field name (character data)	**Field ID**	**Flag 1**	**Flag 2**	**Field length decimal**	**Default value**	**Type**	
Following is the TME segment of the Data Set Template							
TME	001	00	00	00000000	00		Start of segment fields
ROLEN	002	10	00	00000004	00	Int	Count of role-access specifications
ROLES	003	80	00	00000000	00	Char	Role-access specifications

General Template for the Restructured Database

The general template describes the fields of general resource profiles in a RACF database.

NOT programming interface information

ACL2RSVD AUDITQF
AUDITQS CATEGORY
CURKEY CURKEYV
ENCTYPE
FIELD
FLDCNT
FLDFLAG
FLDNAME
FLDVALUE
GAUDITQF
GAUDITQS
MEMCNT
MEMLIST
NUMCTGY
PREVKEY
PREVKEYV
RACLDSP
RACLHDR
SALT
SSKEY

End of NOT programming interface information

Note: Application developers should not depend on being able to use RACROUTE REQUEST=EXTRACT for the BASE segment fields on any security product other than RACF. These products are expected to support only such segments as DFP and TSO.

The contents of the general template are as follows:

Template							**Field being described**
Field name (character data)	**Field ID**	**Flag 1**	**Flag 2**	**Field length decimal**	**Default value**	**Type**	
The following is the BASE segment of the GENERAL template.							
GENERAL	001	00	00	00000000	00		
ENTYPE	002	00	00	00000001	05	Int	The number (5) corresponding to profiles for resources defined in the class descriptor table.
VERSION	003	00	00	00000001	01	Int	The version field from the profile. Always X'01'.
CLASTYPE	004	00	00	00000001	FF	Int	The class to which the resource belongs (from the ID=class-number operand of the ICHERCDE macro).
DEFDATE	005	00	20	00000003	FF	Date	The date the resource was defined to RACF.
OWNER	006	00	00	00000008	FF	Char	The owner of the resource.
LREFDAT	007	01	20	00000003	FF	Date	The date the resource was last referenced.
LCHGDAT	008	01	20	00000003	FF	Date	The date the resource was last updated.
ACSALTR	009	01	00	00000002	FF	Int	The number of times the resource was accessed with ALTER authority.
ACSCNTL	010	01	00	00000002	FF	Int	The number of times the resource was accessed with CONTROL authority.
ACSUPDT	011	01	00	00000002	FF	Int	The number of times the resource was accessed with UPDATE authority.
ACSREAD	012	01	00	00000002	FF	Int	The number of times the resource was accessed with READ authority.

Template							Field being described
Field name (character data)	**Field ID**	**Flag 1**	**Flag 2**	**Field length decimal**	**Default value**	**Type**	
UACC	013	20	80	00000001	00	Bin	The universal access authority for the resource. **Bit** **Meaning when set** **0** ALTER access **1** CONTROL access **2** UPDATE access **3** READ access **4** EXECUTE access **5–6** Reserved for IBM's use **7** NONE access.
AUDIT	014	20	00	00000001	00	Bin	Audit flags. **Bit** **Meaning when set** **0** Audit all accesses **1** Audit successful accesses **2** Audit accesses that fail **3** No auditing **4–7** Reserved for IBM's use
LEVEL	015	20	00	00000001	00	Int	Resource level.
GAUDIT	016	20	00	00000001	00	Bin	Global audit flags. **Bit** **Meaning when set** **0** Audit all accesses **1** Audit successful accesses **2** Audit accesses that fail **3** No auditing **4–7** Reserved for IBM's use
INSTDATA	017	00	00	00000000	00	Char	Installation data; maximum length=255.

Template							Field being described
Field name (character data)	**Field ID**	**Flag 1**	**Flag 2**	**Field length decimal**	**Default value**	**Type**	
GAUDITQF	021	00	00	00000001	FF	Bin	Global audit FAILURES qualifier. The AUDITQS, AUDITQF, GAUDITQS, and GAUDITQF fields have the following format: **Value** **Meaning** **X'00'** Log access at READ authority **X'01'** Log access at UPDATE authority **X'02'** Log access at CONTROL authority **X'03'** Log access at ALTER authority
AUDITQS	018	00	00	00000001	FF	Bin	Audit SUCCESS qualifier. (Audit options specified by a user with the AUDITOR or group-AUDITOR attribute.) **Bit** **Meaning when set** **0** Audit all accesses **1** Audit successful accesses **2** Audit accesses that fail **3** No auditing **4–7** Reserved for IBM's use
AUDITQF	019	00	00	00000001	FF	Bin	Audit FAILURES qualifier. (Audit options specified by a user with the AUDITOR or group-AUDITOR attribute.) **Bit** **Meaning when set** **0** Audit all accesses **1** Audit successful accesses **2** Audit accesses that fail **3** No auditing **4–7** Reserved for IBM's use

Template							Field being described
Field name (character data)	**Field ID**	**Flag 1**	**Flag 2**	**Field length decimal**	**Default value**	**Type**	
GAUDITQS	020	00	00	00000001	FF	Bin	Global audit SUCCESS qualifier. (Audit options specified by a user with the AUDITOR or group-AUDITOR attribute.) **Bit** **Meaning when set** **0** Audit all accesses **1** Audit successful accesses **2** Audit accesses that fail **3** No auditing **4–7** Reserved for IBM's use
WARNING	022	20	00	00000001	00	Bin	Identifies the data set as having (bit 7 is on) or not having the WARNING attribute.
RESFLG	023	20	00	00000001	00	Bin	Resource profile flags: **Bit** **Meaning when set** **0** TAPEVOL can only contain one data set. **1** TAPEVOL profile is automatic. **2** Maintain TVTOC for TAPEVOL. **3–7** Reserved for IBM's use
TVTOCCNT	024	10	00	00000004	00	Int	The number of TVTOC entries.
TVTOCSEQ	025	80	00	00000002	00	Int	The file sequence number of tape data set.
TVTOCCRD	026	80	20	00000003	00	Date	The date the data set was created.
TVTOCIND	027	A0	00	00000001	00	Bin	Data set profiles flag (RACF indicator bit): **Bit** **Meaning when set** **1** Discrete data set profile exists **2–7** Reserved for IBM's use
TVTOCDSN	028	80	00	00000000	00	Char	The RACF internal name.
TVTOCVOL	029	80	00	00000000	00	Char	This field is a list of the volumes on which the tape data set resides.
TVTOCRDS	030	80	00	00000000	00	Char	The name used when creating the tape data set; maximum length=255.
NOTIFY	031	00	00	00000000	00	Char	The user to be notified when access violations occur against resource protected by this profile.
LOGDAYS	032	20	00	00000001	00	Bin	The days of the week the TERMINAL cannot be used. (Bit 0 equals Sunday, bit 1 equals Monday, and so on).
LOGTIME	033	00	00	00000000	00	Time	The time of the day the TERMINAL can be used.
LOGZONE	034	00	00	00000000	00	Bin	The time zone in which the terminal is located.
NUMCTGY	035	10	00	00000004	00	Int	Number of categories.

Template							Field being described
Field name (character data)	**Field ID**	**Flag 1**	**Flag 2**	**Field length decimal**	**Default value**	**Type**	
CATEGORY	036	80	00	00000002	00	Int	List of categories.
SECLEVEL	037	00	00	00000001	FF	Int	Resource security level.
FLDCNT	038	10	00	00000004	00	Int	Reserved for IBM's use.
FLDNAME	039	80	00	00000008	00		Reserved for IBM's use.
FLDVALUE	040	80	00	00000000	00		Reserved for IBM's use.
FLDFLAG	041	A0	00	00000001	00		Reserved for IBM's use.
APPLDATA	042	00	00	00000000	00	Char	Application data.
MEMCNT	043	10	80	00000004	00	Int	The number of members.
MEMLST	044	80	80	00000000	00	Bin	The resource group member. For SECLABEL class, a 4-byte SMF ID.
VOLCNT	045	10	00	00000004	00	Int	Number of volumes in tape volume set.
VOLSER	046	80	00	00000006	00	Char	Volume serials of volumes in tape volume set.
ACLCNT	047	10	80	00000004	00	Int	The number of users and groups currently authorized to access the resource.
USERID	048	80	80	00000008	00	Char	The user ID or group name of each user or group authorized to access the resource.
USERACS	049	A0	80	00000001	00	Bin	The access authority that each user or group has for the resource. **Bit** **Meaning when set** **0** ALTER access **1** CONTROL access **2** UPDATE access **3** READ access **4** EXECUTE access **5–6** Reserved for IBM's use **7** NONE access **Note:** Each of the above access authority fields have mutually-exclusive bits with the exception of EXECUTE+NONE.
ACSCNT	050	80	00	00000002	00	Int	The number of times the resource was accessed by each user or group.
USRCNT	051	10	00	00000004	00	Int	Reserved for installation's use.
USRNM	052	80	00	00000008	00		Reserved for installation's use.
USRDATA	053	80	00	00000000	00		Reserved for installation's use.
USRFLG	054	A0	00	00000001	00		Reserved for installation's use.
SECLABEL	055	00	00	00000008	00	Char	Security label.
ACL2CNT	056	10	00	00000004	00	Int	Number of entries in conditional access list.
ACL2NAME	057	80	00	00000008	00	Bin	1 indicator byte; 7 bytes reserved for IBM's use.
ACL2UID	058	80	00	00000008	00	Char	User ID or group.
ACL2ACC	059	80	00	00000001	00	Bin	Access authority.

Template							Field being described
Field name (character data)	**Field ID**	**Flag 1**	**Flag 2**	**Field length decimal**	**Default value**	**Type**	
ACL2ACNT	060	80	00	00000002	00	Int	Access count.
ACL2RSVD	061	80	00	00000000	00	Bin	Conditional data. Reserved for IBM's use.
RACLHDR	062	00	00	00000020	00	Bin	RACGLIST header.
RACLDSP	063	00	00	00000000	00	Bin	RACGLIST dataspace information.
FILTERCT	064	10	00	00000004	00		Number of names that Hash to this DIGTNMAP Profile.
FLTRLABL	065	80	00	00000000	00		Label associated with this DIGTNMAP Mapping (matches NMAPLABL for user named by FLTRUSER or user irrmulti.)
FLTRSTAT	066	A0	00	00000001	00		Trust status – bit 0 on for trusted.
FLTRUSER	067	80	00	00000000	00		User ID or criteria profile name.
FLTRNAME	068	80	00	00000000	00		Unhashed issuer's name filter used to create this profile name, (max of 255), followed by a separator, (X'4A'), and the unhashed subject's name filter used to create this profile name, (max of 255).
FLTRSVD1	069	80	00	00000000	00		Reserved for IBM's use.
FLTRSVD2	070	80	00	00000000	00		Reserved for IBM's use.
FLTRSVD3	071	80	00	00000000	00		Reserved for IBM's use.
FLTRSVD4	072	80	00	00000000	00		Reserved for IBM's use.
FLTRSVD5	073	80	00	00000000	00		Reserved for IBM's use.
RACDHDR	074	00	08	00000000	00	Bin	CACHECLS header.

Field name	**Field ID**	**Flag 1**	**Flag 2**	**Combination field IDs**					**Type**	
Following is the COMBINATION segment of the GENERAL template.										
CREADATE	000	40	00	005	000	000	000	000		Combination.
AUTHDATE	000	40	00	005	000	000	000	000		Fields.
AUTHOR	000	40	00	006	000	000	000	000		
TVTOC	000	48	00	025	026	027	028	029		
	000	40	00	030	000	000	000	000		
LOGINFO	000	40	00	032	033	034	000	000		
FIELD	000	40	00	039	040	041	000	000		
ACL	000	40	00	048	049	050	000	000		
ACL1	000	40	00	048	049	000	000	000		
USERDATA	000	40	00	052	053	054	000	000		
ACL2	000	40	00	057	058	059	060	061		Conditional access list
ACL2A3	000	40	00	057	058	059	060	000		Conditional access list
FLTRLST1	000	40	00	065	066	067	068	000		Combo field for FILTER
FLTRLST2	000	40	00	065	067	068	000	000		Combo field for FILTER
CERTRING	000	40	00	010	011	009	000	000		Digital Certificate Data.
CERTRNG2	000	40	00	009	011	000	000	000		
CERTRNG3	000	40	00	009	012	013	000	000		

Template							Field being described
Field name (character data)	**Field ID**	**Flag 1**	**Flag 2**	**Field length decimal**	**Default value**	**Type**	
Following is the SESSION segment of the GENERAL template.							
SESSION	001	00	00	00000000	00		Start of segment fields
SESSKEY	002	00	00	00000000	00	Bin	Session key; maximum length = 8
SLSFLAGS	003	20	00	00000001	00	Bin	Session flag byte **Bit** **Meaning when set** **0** SLSLOCK—This profile is locked out **1–7** Reserved for IBM's use
KEYDATE	004	00	00	00000004	00	Date	Last date session key was changed. It is in the format *0cyyddF* where *c*=0 for 1900–1999 and *c*=1 for 2000–2099. For more information on this MVS-returned format, see *z/OS MVS Programming: Assembler Services Guide* .
KEYINTVL	005	00	00	00000002	00	Int	Number of days before session key expires
SLSFAIL	006	00	00	00000002	00	Int	Current number of invalid attempts
MAXFAIL	007	00	00	00000002	00	Int	Number of invalid attempts before lockout
SENTCNT	008	10	00	00000004	00	Int	Number of session entities in list
SENTITY	009	80	00	00000035	00	Char	Entity name
SENTFLCT	010	80	00	00000002	00	Int	Number of failed attempts for this entity
CONVSEC	011	20	00	00000001	00	Bin	Conversation security. **Value** **Meaning** **X'40'** Conversation security **X'50'** Persistent verification **X'60'** User ID and password already verified **X'70'** User ID and password already verified plus persistent verification **X'80'** Security none
Following is the DLFDATA segment of the GENERAL template.							
DLFDATA	001	00	00	00000000	00		Start of segment fields
RETAIN	002	20	00	00000001	00	Bin	Retain flag byte
JOBNMCNT	003	10	00	00000004	00	Int	Count of jobnames
JOBNAMES	004	80	00	00000000	00	Char	Jobnames; maximum length=8
Following is the SSIGNON segment of the GENERAL template.							
SSIGNON	001	00	00	00000000	00		Start of segment fields
SSKEY	002	00	00	00000000	00	Bin	Secured signon key
Following is the STDATA segment of the GENERAL template.							
STDATA	001	00	00	00000000	00		Start of segment fields
STUSER	002	00	00	00000008	40	Char	User ID or =MEMBER
STGROUP	003	00	00	00000008	40	Char	Group name or =MEMBER

Template							Field being described
Field name (character data)	**Field ID**	**Flag 1**	**Flag 2**	**Field length decimal**	**Default value**	**Type**	
FLAGTRUS	004	20	00	00000001	00	Bin	Trusted flag, X'80' = trusted
FLAGPRIV	005	20	00	00000001	00	Bin	Privileged flag, X'80' = privileged
FLAGTRAC	006	20	00	00000001	00	Bin	Trace usage flag X'80' = issue IRR8I2I
Following is the SVFMR segment of the GENERAL template.							
SVFMR	001	00	00	00000000	00		Start of segment fields
SCRIPTN	002	00	00	00000008	00	Char	Script name
PARMN	003	00	00	00000008	00	Char	Parameter name
Following is the CERTDATA segment of the GENERAL template.							
CERTDATA	001	00	00	00000000	00		Start of segment fields
CERT	002	00	00	00000000	00	Bin	Digital Certificate
CERTPRVK	003	00	00	00000000	00	Bin	Private key or key label
RINGCT	004	10	00	00000004	00	Int	Number of key rings associated with this certificate
RINGNAME	005	80	00	00000000	00	Char	Profile name of a ring with which this certificate is associated
CERTSTRT	006	00	00	00000000	00		Date and time from which the certificate is valid. This is an 8-byte TOD format field.
CERTEND	007	00	00	00000000	00		Date and time after which the certificate is not valid. This is an 8-byte TOD format field.
							The CERTCT repeat group identifies the certificates that are associated with a key ring. It is used **only** with DIGTRING profiles.
CERTCT	008	10	00	00000004	00	Int	The number of certificates associated with this key ring
CERTNAME	009	80	00	00000000	00	Char	The profile name of the certificate
CERTUSAG	010	80	00	00000004	00	Bin	Certificate usage in ring: • X'00000000' – PERSONAL • X'00000001' – SITE • X'00000002' – CERTAUTH
CERTDFLT	011	80	00	00000001	00	Bin	Verifies if it is the default certificate: • X'00' – Not the default • X'80' – The default
CERTSJDN	012	80	00	00000000	00	Bin	The subject name of the entity to whom the certificate is issued. This field is a BER-encoded format of the subject's distinguished name as contained in the certificate
CERTLABL	013	80	00	00000000	00	Char	Label associated with the certificate
CERTRSV1	014	80	00	00000000	00		Reserved for IBM's use.
CERTRSV2	015	80	00	00000000	00		Reserved for IBM's use.
CERTRSV3	016	80	00	00000000	00		Reserved for IBM's use.
CERTRSV4	017	80	00	00000000	00		Reserved for IBM's use.
CERTRSV5	018	80	00	00000000	00		Reserved for IBM's use.
CERTRSV6	019	80	00	00000000	00		Reserved for IBM's use.
CERTRSV7	020	80	00	00000000	00		Reserved for IBM's use.
CERTRSV8	021	80	00	00000000	00		Reserved for IBM's use.

Template							Field being described
Field name (character data)	**Field ID**	**Flag 1**	**Flag 2**	**Field length decimal**	**Default value**	**Type**	
CERTRSV9	022	80	00	00000000	00		Reserved for IBM's use.
CERTRSVA	023	80	00	00000000	00		Reserved for IBM's use.
CERTRSVB	024	80	00	00000000	00		Reserved for IBM's use.
CERTRSVC	025	80	00	00000000	00		Reserved for IBM's use.
CERTRSVD	026	80	00	00000000	00		Reserved for IBM's use.
CERTRSVE	027	80	00	00000000	00		Reserved for IBM's use.
CERTRSVF	028	80	00	00000000	00		Reserved for IBM's use.
CERTRSVG	029	80	00	00000000	00		Reserved for IBM's use.
CERTRSVH	030	80	00	00000000	00		Reserved for IBM's use.
CERTRSVI	031	80	00	00000000	00		Reserved for IBM's use.
CERTRSVJ	032	80	00	00000000	00		Reserved for IBM's use.
CERTRSVK	033	80	00	00000000	00		Reserved for IBM's use.
CERTPRVT	034	00	00	00000004	00	Bin	Associated key type: • X'00000000' – No associated key, • X'00000001' – PKCS DER-encoded, • X'00000002'> – ICSF token label, • X'00000003' – PCICC label, • X'00000004' – DSA, • >X'00000005' – ICSF public token label
CERTPRVS	035	00	00	00000004	00	Int	Private key size in bits
CERTLSER	036	00	00	00000008	00	Bin	The low order 8 bytes of the last certificate that was signed with this key. This field is used with DIGTCERT profiles only
RINGSEQN	037	00	00	00000004	00	Int	Ring change count
Following is the TME segment of the GENERAL template.							
TME	001	00	00	00000000	00		Start of segment fields
PARENT	002	00	00	00000000	00	Char	Parent name
CHILDN	003	10	00	00000004	00	Int	Count of children
CHILDREN	004	80	00	00000000	00	Char	Child names
RESN	005	10	00	00000004	00	Int	Count of resource-access specifications
RESOURCE	006	80	00	00000000	00		Resource-access specifications
GROUPN	007	10	00	00000004	00	Int	Count of groups
GROUPS	008	80	00	00000008	00		Group names
ROLEN	009	10	00	00000004	00	Int	Count of role-access specifications
ROLES	010	80	00	00000000	00	Char	Role-access specifications
Following is the KERB segment of the GENERAL template							
KERB	001	00	00	00000000	00		Start of segment fields
KERBNAME	002	00	00	00000000	00	Char	Kerberos realm name
MINTKTLF	003	00	00	00000000	00	Char	Minimum ticket life
MAXTKTLF	004	00	00	00000000	00	Char	Maximum ticket life
DEFTKTLF	005	00	00	00000000	00	Char	Default ticket life
SALT	006	00	00	00000000	00	Char	Current key salt

Template							Field being described
Field name (character data)	**Field ID**	**Flag 1**	**Flag 2**	**Field length decimal**	**Default value**	**Type**	
ENCTYPE	007	00	00	00000000	00	Char	Encryption type
CURKEYV	008	00	00	00000000	00	Char	Current key version
CURKEY	009	00	00	00000000	00	Char	Current DES key
PREVKEYV	010	00	00	00000000	00	Char	Previous key version
PREVKEY	011	00	00	00000000	00	Char	Previous DES key
ENCRYPT	012	00	00	00000004	55	Char	Encryption types for SETROPTS KERBLVL(1)
Following is the PROXY segment of the GENERAL template							
PROXY	001	00	00	00000000	00		Start of segment fields
LDAPHOST	002	00	00	00000000	00	Char	LDAP server URL; maximum length: 1023
BINDDN	003	00	00	00000000	00	Char	Bind distinguished name; maximum length: 1023
BINDPW	004	00	08	00000000	00	Char	Bind password; maximum length: 128
BINDPWKY	005	00	08	00000071	00	Char	Bind password mask or encrypt key
Following is the EIM segment of the GENERAL template							
EIM	001	00	00	00000000	00		Start of segment fields
DOMAINDN	002	00	00	00000000	00	Char	EIM Domain Distinguished Names
OPTIONS	003	00	00	00000004	55	Char	EIM Options
LOCALREG	004	00	00	00000000	00	Char	Local Registry Name
KERBREG	005	00	00	00000000	00	Char	Kerberos Registry Name
X509REG	006	00	00	00000000	00	Char	X509 Registry Name
Following is the ALIAS segment of the GENERAL template							
ALIAS	001	00	00	00000000	00		Start of segment fields
IPLOOK	002	00	10	00000016	00	Bin	IP lookup value
Following is the CDTINFO segment of the GENERAL template							
CDTINFO	001	00	00	0	0		Start of segment fields
CDTPOSIT	002	00	00	4	FF	Int	POSIT number for class
CDTMAXLN	003	00	00	1	8	Int	Maximum length of profile names
CDTMAXLX	004	00	00	4	FF	Int	Maximum resource or profile name length when using ENTITYX
CDTDFTRC	005	00	00	1	4	Int	Default return code
CDTKEYQL	006	00	00	4	0	Int	Number of key qualifiers
CDTGROUP	007	00	00	8	0	Char	Resource grouping class name
CDTMEMBR	008	00	00	8	0	Char	Member class name

Template							Field being described
Field name (character data)	**Field ID**	**Flag 1**	**Flag 2**	**Field length decimal**	**Default value**	**Type**	
CDTFIRST	009	00	00	1	X'C0'	Bin	Character restriction for first character of profile name **Value** **Meaning** **X'80'** Alphabetic **X'40'** National **X'20'** Numeric **X'10'** Special
CDTOTHER	010	00	00	1	X'C0'	Bin	Character restriction for characters of the profile name other than the first character **Value** **Meaning** **X'80'** Alphabetic **X'40'** National **X'20'** Numeric **X'10'** Special
CDTOPER	011	00	00	1	X'00'	Bin	Operations attribute considered **Value** **Meaning** **X'80'** RACF considers OPERATIONS attribute
CDTUACC	012	00	00	1	X'01'	Bin	Default UACC **Value** **Meaning** **X'80'** ALTER **X'40'** CONTROL **X'20'** UPDATE **X'10'** READ **X'08'** EXECUTE **X'04'** UACC from ACEE **X'01'** NONE

Template							Field being described
Field name (character data)	**Field ID**	**Flag 1**	**Flag 2**	**Field length decimal**	**Default value**	**Type**	
CDTRACL	013	00	00	1	X'00'	Bin	SETROPTS RACLIST **Value** **Meaning** **X'00'** RACLIST disallowed **X'80'** RACLIST allowed **X'40'** RACLIST required
CDGENL	014	00	00	1	X'00'	Bin	SETROPTS GENLIST **Value** **Meaning** **X'80'** GENLIST allowed
CDTPRFAL	015	00	00	1	X'80'	Bin	Profiles allowed **Value** **Meaning** **X'80'** Profiles are allowed
CDTSLREQ	016	00	00	1	X'00'	Bin	Security labels required **Value** **Meaning** **X'80'** Security labels are required
CDTMAC	017	00	00	1	X'80'	Bin	Mandatory access checking (MAC) processing **Value** **Meaning** **X'80'** Normal mandatory access checks **X'40'** Reverse mandatory access checks **X'20'** Equal mandatory access checks
CDTSIGL	018	00	00	1	X'00'	Bin	ENF Signal **Value** **Meaning** **X'80'** ENF signal to be sent
CDTCASE	019	00	00	1	X'00'	Bin	Case of profile names **Value** **Meaning** **X'00'** Upper case **X'80'** ASIS – preserve case
CDTGEN	020	00	00	1	X'80'	Bin	SETROPTS GENERIC **Value** **Meaning** **X'80'** GENERIC allowed

Template							Field being described
Field name (character data)	**Field ID**	**Flag 1**	**Flag 2**	**Field length decimal**	**Default value**	**Type**	
Following is the ICTX segment of the GENERAL template							
ICTX	001	00	00	00000000	00		Start of segment fields
USEMAP	002	00	00	00000001	80	Bin	Application supplied mapping **Value** **Meaning** **X'80'** Use the mapping
DOMAP	003	00	00	00000001	00	Bin	Identity cache mapping **Value** **Meaning** **X'80'** Do the mapping
MAPREQ	004	00	00	00000001	00	Bin	**Value** **Meaning** **X'80'** Mapping is required
MAPTIMEO	005	00	00	00000002	00	Int	Mapping timeout adjustment

Reserved Templates for the Restructured Database

Five unused templates are defined for future use. The installation must leave this space reserved and not use it.

The contents of the reserved template are as follows:

Template							Field being described
Field name (character data)	**Field ID**	**Flag 1**	**Flag 2**	**Field length decimal**	**Default value**	**Type**	
RSVTMP03	001	00	00	00000000	00		
ENTYPE	002	00	00	00000001	00		The number corresponding to the type of profile being described.
VERSION	003	00	00	00000001	00		Template version number.

Appendix G. Contents of the encrypted password or password phrase envelope

When the enveloping function is enabled for passwords or password phrases, an encrypted copy of the password or phrase can be retrieved using LDAP and the SDBM backend, by retrieving the racfPasswordEnvelope or racfPassPhraseEnvelope attribute, respectively. This is the only supported method of retrieving the envelope. RACROUTE REQUEST=EXTRACT will return an internal format of the envelope which cannot be decrypted.

The PKCS #7 standard defines the structure of encrypted content comprising a digital envelope which is intended for multiple recipients. RACF will first sign, and then envelope (encrypt) the password or password phrase payload (defined below). The recipient will decrypt the envelope to obtain the payload. The recipient should also verify the signature of the envelope, although this is not necessary in order to obtain the payload.

Note that a digital certificate is required to verify the signature of the envelope. See the description of the RACF password and password phrase enveloping function in *z/VM: RACF Security Server Security Administrator's Guide* on how the recipient can obtain the digital certificate.

For the structure of the envelope, refer to the PKCS #7 standard, Version 1.5.

PKCS #7 defines several data types using ASN.1 notation to describe them. Each type is contained in a ContentInfo type. ContentInfo simply identifies the contained data type with an object identifier (OID), followed by the actual data. See section 7 of the PKCS #7 standard.

On the outside of the structure is a ContentInfo containing the EnvelopedData content type. The EnvelopedData type is described in section 10 of the PKCS #7 standard. The EnvelopedData type is further broken down into subtypes such as RecipientInfos and EncryptedContentInfo. EncryptedContentInfo is broken down further into ContentType, ContentEncryptionAlgorithmIdentifier, and EncryptedContent. ContentType will be SignedData, and EncryptedContent, defined in the standard as OCTET STRING (which just means an arbitray string of data), will be the ContentInfo containing data of type SignedData which is the output of the System SSL signing function. See section 9.1 of the PKCS #7 standard describing the SignedData type.

SignedData contains a field called contentInfo, which is the data being signed. This data can be of any of the types defined by the standard. The type we use is Data, so the contentInfo field within SignedData contains the ContentInfo for the Data type. The Data type is defined as OCTET STRING. This OCTET STRING is the payload that RACF is constructing as input to the whole envelope-creation process. The payload contains password related information in BER-encoded ASCII format.

The following is the ASN.1 notation describing the password payload as constructed by RACF:

```
PasswordPayload::=SEQUENCE{
                    Version                         INTEGER
                    Expired                         BOOLEAN
                    Password                        UTF8String
                    Changetime                      IA5String
                    Language                        IA5String OPTIONAL   DEFAULT "ENU"
}
```

The following is the ASN.1 notation describing the password phrase payload as constructed by RACF:

```
PasswordphrasePayload::=SEQUENCE{
                        Version                 INTEGER
                        Expired                 BOOLEAN
                        Passwordphrase          UTF8String
                        Changetime              IA5String
                        Language                IA5String OPTIONAL DEFAULT "ENU"
}
```

Version is the version number of the payload. For the password payload, it is set to 1 if the password has been changed to lowercase, or 2 if the password appears as entered. For the password phrase payload, it is set to 1.

Expired will be true if the new password or password phrase is marked as expired at the time of the change (for example, an ALTUSER command is used to change the password or password phrase without specifying the NOEXPIRED operand). If Expired is true, the password or password phrase must be changed the next time the user logs on.

Password is the value of the new password. If the mixed case password support is not active, the password is in lowercase. If it is active, the password case is as entered.

Passwordphrase is the value of the new password phrase, with case preserved.

Changetime is a character string of decimal numbers in the format *yyyymmddhhiiss.uuuuuuZ* (relative to GMT) where

- *yyyy* is year
- *mm* is month
- *dd* is day
- *hh* is hour
- *ii* is minutes
- *ss* is seconds
- *uuuuuu* is micro seconds
- 'Z' is a character constant meaning that this time is based on ZULU time, also known as GMT

Language is the 3 character language code which RACF has used in order to determine the UTF-8 code points for the variant characters. This is for diagnostic purposes. Currently, RACF assumes the language is U.S. English ('ENU'). This may result in RACF propagating a different password or password phrase than may be expected by a given user using a given keyboard and code page. If so, users should avoid using variant characters in passwords and password phrases when RACF is participating in a password synchronization network. For example, a person in the United Kingdom may enter the pound sterling symbol as a character in a new password. This is represented as X'5B' which RACF will accept. When RACF envelopes this password assuming U.S. English, the UTF-8 code point for '$' will be used. If this password is propagated to another system, and the person tries to log on to that system using the same keystrokes he used to change his password in RACF, the password will be rejected.

Appendix H. Event Code Qualifiers

The RACF event code (found in the SMF80EVT field of the SMF record) and the RACF event code qualifier (found in the SMF80EVQ field of the SMF record) are determined during RACF processing. The following sections explain the meaning of each qualifier code by event. Some of these event codes and qualifiers apply only to z/OS systems, but are listed here for completeness.

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

This event is logged by RACROUTE REQUEST=VERIFY and RACROUTE REQUEST=VERIFYX.

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

0(0)
SUCCESSFUL INITIATION The job began successfully.

1(1)
INVALID PASSWORD The password specified on the job card or at logon is incorrect.

2(2)
INVALID GROUP The user tried to log on or to initiate a job using a group that the user is not a member of.

3(3)
INVALID OIDCARD Operator identification cards are used at the installation, and the data received from the one used does not match that of the user's profile.

4(4)
INVALID TERMINAL/CONSOLE The user is not authorized to the port of entry (POE). There are four kinds of POEs, each with its own profile class: APPCPORT, CONSOLE, JESINPUT, and TERMINAL. One of the following occurred:

- The port of entry is active but the user is not authorized.
- The user is denied access because of conditional days/times in the user profile.
- The user is denied access because of conditional days/times in the class profile (TERMINAL class only).

5(5)
INVALID APPLICATION The APPL class is active, and the user is trying to log on to an application without authorization.

6(6)
REVOKED USER ID ATTEMPTING ACCESS The user ID specified on the logon has been revoked. One of the following occurred:

- The installation-defined limit of password attempts was reached at an earlier time.
- The inactive interval was reached.
- The revoke-date in the user's profile is in effect.
- The RACF administrator revoked the user ID.

The RACF administrator must reset the user ID before the user can log on again.

7(7)
USER ID AUTOMATICALLY REVOKED The user ID has been automatically revoked. The installation-defined limit of password and password phrase attempts was reached.

8(8)
SUCCESSFUL TERMINATION The job completed successfully.

9(9)
UNDEFINED USER ID The user ID specified on the job card or at logon is not defined to the RACF database.

10(A)
INSUFFICIENT SECURITY LABEL AUTHORITY One of the following occurred:

- SETROPTS MLS FAILURES is in effect and the user's security label does not dominate the submitter's security label. Two exceptions are explained under Qualifier 20.
- SETROPTS MLACTIVE FAILURES is in effect and the job card/logon attempt does not specify a valid security label. One exception is explained under Qualifier 21.

11(B)
NOT AUTHORIZED TO SECURITY LABEL The user is not authorized to the security label specified. One exception is explained under Qualifier 22.

12(C)
SUCCESSFUL RACINIT INITIATION The job or user was verified.

13(D)
SUCCESSFUL RACINIT DELETE The job completed or the user logged off.

14(E)
SYSTEM NOW REQUIRES MORE AUTHORITY SETROPTS MLQUIET is in effect. If this is a user verification, the user is not a console operator and does not have the SPECIAL attribute. If this is a job verification, the job is not part of the trusted computing base (TCB). The verification fails.

15(F)
REMOTE JOB ENTRY—JOB NOT AUTHORIZED The submitting node is not authorized to the system; a NODES profile prevents remote job entry. The profile has the format `'submit_node.RUSER.userid'` and has a UACC of NONE.

16(10)
SURROGATE CLASS IS INACTIVE The SURROGAT class is inactive. The job card has a user ID that is different from the submitter's user ID, and there is no password specified. On VM, someone attempted to logon to a shared user ID.

17(11)
SUBMITTER IS NOT AUTHORIZED BY USER The SURROGAT class is active. Either there is no SURROGAT profile for the job card's user ID or the submitter's user ID is not permitted to the profile. On VM, someone attempted to logon to a shared user ID. Either there is no SURROGAT profile for the shared user ID or the user logging on to the shared user ID is not permitted to the SURROGAT profile.

18(12)
SUBMITTER IS NOT AUTHORIZED TO SECURITY LABEL The SECLABEL class is active and there is a security label on the job card. The submitter is not authorized to the security label specified on the job card.

19(13)
USER IS NOT AUTHORIZED TO JOB The JESJOBS class is active, and the user is not authorized to the jobname.

20(14)
WARNING—INSUFFICIENT SECURITY LABEL AUTHORITY One of the following occurred:

- SETROPTS MLS WARNING is in effect and the security label on the job card does not dominate the submitter's security label.
- SETROPTS MLS FAILURES is in effect, the user's security label does not dominate the submitter's, and the user has the SPECIAL attribute.
- SETROPTS MLS FAILURES and SETROPTS COMPATMODE are in effect, the user's security label does not dominate the submitter's, and the submitter's or the job owner's security label is the default.

The verification does not fail.

21(15)

WARNING—SECURITY LABEL MISSING FROM JOB, USER, OR PROFILE One of the following occurred:

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

The verification does not fail.

22(16)

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

23(17)

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

One exception is listed under Qualifier 24.

24(18)

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

25(19)

CURRENT PASSWORD HAS EXPIRED The user's password has expired for one of the following reasons:

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

26(1A)

INVALID NEW PASSWORD The new password specified may be incorrect because:

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

27(1B)

VERIFICATION FAILED BY INSTALLATION The installation exit ICHRIX01 failed the request.

28(1C)

GROUP ACCESS HAS BEEN REVOKED The user's membership to the group specified has been revoked.

29(1D)

OIDCARD IS REQUIRED An OIDCARD is required by the installation but none was given.

30(1E)

NETWORK JOB ENTRY—JOB NOT AUTHORIZED For session types of NJE SYSOUT or NJE BATCH, the verification fails because one of the following occurred:

- The user, group, or security label requirements in the NODES profiles were not met.
- The submitter's node is not valid.
- The reverify check failed.

See *z/VM: RACF Security Server System Programmer's Guide* for details on NJE.

31(1F)
WARNING—UNKNOWN USER FROM TRUSTED NODE PROPAGATED The combination of having a trusted node submit a job with the undefined user ID warrants this logging. The verification does not fail.

For an NJE BATCH job, the submitting user is the NJE undefined user ID. The default NJE undefined user ID is eight question marks (????????), unless it was changed with the SETROPTS JES NJEUSERID command. The submitting node is trusted (its best-fit NODES profile on the receiving node's system has a UACC of at least UPDATE). This profile allows propagation of submitters; however, the undefined user ID does not propagate.

32(20)
SUCCESSFUL INITIATION USING PASSTICKET Logon was achieved using a PassTicket.

33(21)
ATTEMPTED REPLY OF PASSTICKET Logon was rejected because of attempted replay of a PassTicket.

35(23)
USER AUTOMATICALLY REVOKED DUE TO INACTIVITY A user has not logged on or accessed the system for so long that the user ID has become inactive. RACF prevents the user from accessing the system.

36(24)
PASS PHRASE IS NOT VALID A user attempted to access the system specifying a password phrase that is not valid. RACF prevents the user from accessing the system.

37(25)
NEW PASS PHRASE IS NOT VALID Logon was rejected because the new password phrase is not valid.

38(26)
CURRENT PASS PHRASE HAS EXPIRED Logon was rejected because the current password phrase has expired.

40(28)
SUCCESSFUL MULTI-FACTOR AUTHENTICATION Multi-Factor Authentication was successful.

41(29)
FAILED MULTI-FACTOR AUTHENTICATION Multi-Factor Authentication failed.

42(2A)
MULTI-FACTOR AUTHENTICATION UNAVAILABLE Multi-Factor Authentication was unavailable.

Event 2(2): RESOURCE ACCESS

This event is logged by RACROUTE REQUEST=AUTH.

This event is also logged by RACROUTE REQUEST=FASTAUTH if auditing the PROGRAM class. Only qualifiers 0, 1, and 3 are used by RACROUTE REQUEST=FASTAUTH.

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

0(0)
SUCCESSFUL ACCESS The user has authorization to the resource.

1(1)
INSUFFICIENT AUTHORITY The user does not have authorization to the resource.

2(2)

PROFILE NOT FOUND—RACFIND SPECIFIED ON MACRO If the request is AUTH, the RACFIND keyword equaled YES on the authorization request, specifying that a discrete profile should exist for the resource. No discrete or generic RACF protection was found.

If the request is FASTAUTH, the program is not controlled and the PADS data sets are open.

3(3)

ACCESS PERMITTED DUE TO WARNING The user does not have proper authority to the resource. However, the resource's profile has the WARNING option and allows the access.

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

4(4)

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

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

5(5)

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

The exceptions in Qualifier 4 also apply.

6(6)

INSUFFICIENT CATEGORY/SECLEVEL The installation uses categories or security levels as separate entities. One of the following occurred:

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

7(7)

INSUFFICIENT SECURITY LABEL AUTHORITY The SECLABEL class is active and one of the following occurred:

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

8(8)

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

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

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

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

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

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

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

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

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

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

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

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

Exceptions:
• PROGRAM class profiles cannot use the WARNING option. • RACLISTed profiles can use the WARNING option only if they are RACLISTed by SETROPTS or a RACF 1.8 (or later) RACROUTE REQUEST=LIST.

Event 3(3): ADDVOL/CHGVOL

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

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

0(0)
SUCCESSFUL PROCESSING OF NEW VOLUME One of the following occurred:

- The user has proper administrative authority to the DATASET profile; in the case of tape data sets with TAPEVOL active, the user also had administrative authority to the TAPEVOL profile.
- SETROPTS MLS WARNING is in effect, the TAPEVOL class is active, a TAPEVOL profile exists, and the user's security label does not equal the resource's.
- SETROPTS MLACTIVE WARNING is in effect, the TAPEVOL class is active, and no TAPEVOL profile exists for the volume.

1(1)

INSUFFICIENT AUTHORITY The user did not have administrative authority to the DATASET profile, or, in the case of tape data sets, the TAPEVOL class is active and the user did not have administrative authority to the TAPEVOL profile.

2(2)

INSUFFICIENT SECURITY LABEL AUTHORITY The SECLABEL class is active, the data set is a tape data set, the TAPEVOL class is active, and the user's security label does not dominate the security label found in the TAPEVOL profile.

3(3)

LESS SPECIFIC PROFILE EXISTS WITH DIFFERENT SECLABEL The SECLABEL class is active, SETROPTS MLSTABLE is in effect, a less specific generic profile exists that does not have the same security label, the data set is a tape data set, and the TAPEVOL class is active. Changing the volume would change the TAPEVOL profile's security label, violating SETROPTS MLSTABLE rules.

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

Event 4(4): RENAME RESOURCE

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

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

0(0)

SUCCESSFUL RENAME One of the following occurred:

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

1(1)

INVALID GROUP The resource to be renamed is a data set, and the high-level qualifier of the new data set is not a valid group, or user ID.

2(2)

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

3(3)

INSUFFICIENT AUTHORITY One of the following occurred:

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

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

4(4)

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

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

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

6(6)
RESOURCE NOT PROTECTED SETROPTS PROTECTALL FAILURES is in effect, and the new data set name is not protected by a profile.

7(7)
WARNING—RESOURCE NOT PROTECTED SETROPTS PROTECTALL WARNINGS is in effect, and the new data set name is not protected by a profile.

The RENAME is allowed.

8(8)
USER IN SECOND QUALIFIER IS NOT RACF DEFINED The second qualifier of the new name is not a valid user ID.

9(9)
LESS SPECIFIC PROFILE EXISTS WITH DIFFERENT SECLABEL The SECLABEL class is active, SETROPTS MLSTABLE is in effect, and there is a less specific generic profile existing for the new name with a different security label. Renaming this resource would violate SETROPTS MLSTABLE rules.

10(A)
INSUFFICIENT SECURITY LABEL AUTHORITY The SECLABEL class is active, SETROPTS MLS FAILURES is in effect, and the user is not authorized to the security label of the resource to be renamed.

11(B)
RESOURCE NOT PROTECTED BY SECURITY LABEL The SECLABEL class is active, SETROPTS MLS FAILURES is in effect, and the profile covering the old resource name does not have a security label.

12(C)
NEW NAME NOT PROTECTED BY SECURITY LABEL The SECLABEL class is active, SETROPTS MLS FAILURES is in effect, and the profile that would cover the new resource name does not have a security label.

13(D)
NEW SECLABEL MUST DOMINATE OLD SECLABEL The SECLABEL class is active, SETROPTS MLS FAILURES is in effect, and the security label of the profile covering the new resource name does not dominate the security label of the profile covering the old resource name.

14(E)
INSUFFICIENT SECURITY LABEL AUTHORITY The SECLABEL class is active, SETROPTS MLS WARNING is in effect, and the user is not authorized to the security label of the profile. The RENAME is allowed.

15(F)
WARNING—RESOURCE NOT PROTECTED BY SECURITY LABEL The SECLABEL class is active, SETROPTS MLS WARNING is in effect, and the profile covering the old resource name does not have a security label. The RENAME is allowed.

16(10)
WARNING—NEW NAME NOT PROTECTED BY SECURITY LABEL The SECLABEL class is active, SETROPTS MLS WARNING is in effect, and the profile that would cover the new resource name does not have a security label. The RENAME is allowed.

17(11)
WARNING—NEW SECLABEL MUST DOMINATE OLD SECLABEL The SECLABEL class is active, SETROPTS MLS WARNING is in effect, and the security label of the profile covering the new resource name does not dominate the security label of the profile covering the old resource name. The RENAME does not fail.

Event 5(5): DELETE RESOURCE

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

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

0(0)
SUCCESSFUL SCRATCH The resource profile was deleted.

1(1)
RESOURCE NOT FOUND The resource profile was not found.

2(2)
INVALID VOLUME The class is DATASET, and the data set does not reside on the volume specified.

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

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

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

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

Event 7(7): DEFINE RESOURCE

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

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

0(0)
SUCCESSFUL DEFINITION

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

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

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

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

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

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

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

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

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

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

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

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

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

The DEFINE is allowed.

10(A)
USER IN SECOND QUALIFIER IS NOT RACF-DEFINED The second qualifier of the name is not a valid user ID.

11(B)
INSUFFICIENT SECURITY LABEL AUTHORITY The SECLABEL class is active, and one of the following occurred:

- SETROPTS MLACTIVE FAILURES is in effect, and the user is missing a security label.
- SETROPTS MLACTIVE FAILURES is in effect, and the resource is missing a security label.
- The user's security label does not dominate the resource's.
- SETROPTS MLS FAILURES is in effect, and the user's security label does not equal the resource's.

12(C)
LESS SPECIFIC PROFILE EXISTS WITH A DIFFERENT SECLABEL The SECLABEL class is active, SETROPTS MLSTABLE is in effect, and there is a less specific generic profile existing for the name with a different security label.

Defining this resource would violate SETROPTS MLSTABLE rules.

Events 8(8)–25(19): COMMANDS

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

0(0)
NO VIOLATIONS DETECTED The RACF command was issued successfully. This qualifier applies to all RACF commands.

1(1)
INSUFFICIENT AUTHORITY The user did not have the authority to issue the RACF command. This qualifier applies to all RACF commands.

2(2)
KEYWORD VIOLATIONS DETECTED The user had the authority to issue the RACF command, but not to all the keywords that were specified. Keywords that the user is not authorized to use are ignored. For example, a user with the SPECIAL attribute but without the AUDITOR attribute can issue

the ALTUSER command, but not with the GLOBALAUDIT keyword. This qualifier applies to all RACF commands.

3(3)
SUCCESSFUL LISTING OF DATASETS This logs the successful use of LISTDSD DSNS.

4(4)
SYSTEM ERROR IN LISTING OF DATA SETS This logs an error in attempting LISTDSD DSNS.

Event 26(1A): APPCLU

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

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

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

0(0)
PARTNER VERIFICATION WAS SUCCESSFUL The handshaking was successful. The LUs established a connection.

1(1)
SESSION ESTABLISHED WITHOUT VERIFICATION No handshaking was done, but the LUs were still allowed to establish a connection, with the knowledge that the partners were not verified.

2(2)
LOCAL LU KEY WILL EXPIRE IN 5 DAYS OR LESS The handshaking was successful. This qualifier was set to tell users when the local LU's session key would expire.

3(3)
PARTNER LU ACCESS HAS BEEN REVOKED Too many unsuccessful attempts were made at the session key.

4(4)
PARTNER LU KEY DOES NOT MATCH THIS LU KEY An attempt was made, but the session keys did not match; for example, the two sets of identical data encrypted with the two keys did not match.

5(5)
SESSION TERMINATED FOR SECURITY REASONS One or both of the APPCLU profiles involved have the keyword LOCK specified in their session information, preventing any connections from being made. This keyword enables the security administrator to temporarily prevent specific connections without deleting any profiles.

6(6)
REQUIRED SESSION KEY NOT DEFINED The local LU had VERIFY=REQUIRED coded on its APPL statement, indicating that session level verification must be used on all sessions with the LU. One of the following occurred:

- The local LU is the primary LU and no password was defined in RACF for the LU pair.
- The partner LU is the primary LU, but the bind it sent to the local LU did not contain random data (which would indicate that the partner is using session level verification also).

7(7)
POSSIBLE SECURITY ATTACK BY PARTNER LU The local LU sent out a random number to another LU as part of the handshaking process of establishing a session. That same number then came in from a third LU for the local LU to encrypt. It is a coincidence that the same number is chosen; the number is 64 bits of random data.

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

8(8)
SESSION KEY NOT DEFINED FOR PARTNER LU The local LU had VERIFY=OPTIONAL coded on its APPL statement. There was a password defined in the local LU's RACF profile for the LU-LU pair, indicating that session level verification should be used on all sessions between the two LU's. However, the partner LU tried to start a session without using session level verification.

9(9)
SESSION KEY NOT DEFINED FOR THIS LU The local LU had VERIFY=OPTIONAL coded on its APPL statement. No password was defined in the local LU's RACF profile for the LU-LU pair, indicating that session level verification may not be used to establish sessions with this LU. However, the partner LU tried to establish a session using session level verification.

10(A)
SNA SECURITY-RELATED PROTOCOL ERROR The LU trying to establish a connection is not responding correctly according to the handshaking protocol.

11(B)
PROFILE CHANGE DURING VERIFICATION The handshaking was attempted, but it is evident that one of the LU's profiles (specifically the session key) changed in the middle of the handshaking, making its success impossible.

12(C)
EXPIRED SESSION KEY The session key in one or both of the APPCLU profiles has expired.

Event 27(1B): GENERAL AUDITING

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

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

0 - 99 GENERAL AUDIT RECORD WRITTEN
Qualifiers 0 to 99 can be used for Event 27. These qualifiers are installation defined.

Event 28(IC)–56(38): OPENEXTENSIONS EVENT TYPES

Events 28 through 56 apply to OpenExtensions VM. The following qualifier codes are used for each event:

28(1C)
DIRECTORY SEARCH

0(0)
Access allowed

1(1)
Not authorized to search directory

29(1D)
CHECK ACCESS TO DIRECTORY

0(0)
Access allowed

1(1)
Caller does not have requested access authority

30(1E)
CHECK ACCESS TO FILE

0(0)
Access allowed

1(1)
Caller does not have requested access authority

31(1F)
CHAUDIT

0(0)
File's audit options changed

1(1)
Caller does not have authority to change user audit options of specified file

2(2)
Caller does not have authority to change auditor audit options

33(21)
CHMOD

0(0)
File's mode changed

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

34(22)
CHOWN

0(0)
File's owner or group owner changed

1(1)
Caller does not have authority to change owner or group owner of specified file

36(24)
EXEC WITH SETUID/SETGID

0(0)
Successful change of UIDs and GIDs

1(1)
Caller does not have access to the appropriate EXEC.U*uid* profile in the VMPOSIX class.

This qualifier is relevant only to VM.

2(2)
Caller does not have access to the appropriate EXEC.G*gid* profile in the VMPOSIX class.

This qualifier is relevant only to VM.

41(29)
LINK

0(0)
New link created

*****
Failures logged as directory search or check access event types

42(2A)
MKDIR

0(0)
Directory successfully created

*****
Failures logged as directory search or check access event types

43(2B)
MKNOD

0(0)
Successful creation of a node

*****
Failures logged as directory search or check access event types

45(2D)
OPEN (NEW FILE)

0(0)
File successfully created

*****
Failures logged as directory search or check access event types

47(2F)
RENAME

0(0)
Rename successful

*****
Failures logged as directory search or check access event types

48(30)
RMDIR

0(0)
Successful rmdir

*****
Failures logged as directory search or check access event types

49(31)
SETEGID

0(0)
Successful change of effective GID

1(1)
Not authorized to setegid

50(32)
SETEUID

0(0)
Successful change of effective UID

1(1)
Not authorized to seteuid

51(33)
SETGID

0(0)
Successful change of GIDs

1(1)
Not authorized to setgid

52(34)
SETUID

0(0)
Successful change of UIDs

1(1)
Not authorized to setuid

53(35)
SYMLINK

0(0)
Successful symlink

*****
Failures logged as directory search or check access event types

54(36)
UNLINK

0(0)
Successful unlink

***
Failures logged as directory search or check access event types

56(38)
CHECK FILE OWNER

0(0)
User is the owner

1(1)
User is not the owner

Appendix I. OpenExtensions Audit Function Codes

This appendix documents the audit function codes contained in data type 256 for OpenExtensions audit records. The audit function code identifies the OpenExtensions service which triggered the creation of the audit record.

The audit function code appears in the SMF type 80 record as a halfword numeric value shown in the "Value" column. When SMF data unload processes the SMF type 80 record, the halfword value is converted to an 11 character string shown in the "Character String" column. It is important to note that the string value represents the service which was invoked, but does not indicate how it was invoked. For example, the SETEUID (set effective UID) function could have been invoked as the seteuid() C runtime syscall, as the BPX1SEU callable service, or as a DIAGNOSE X'29C'.

Table 191 on page 439 contains the values that are audited in both z/VM and z/OS environments. The audit function codes that are audited on z/VM are a subset of those audited on z/OS.

These definitions are available in IRRPAFC in RACF MACLIB.

Table 191. OpenExtensions Audit Function Codes

Name	Description	Value	Character String
AFC_ACCESS	ck_access	1	ACCESS
AFC_CHAUDIT_U	chg user audit options	2	CHAUDIT
AFC_CHDIR	chg current working directory	3	CHDIR
AFC_CHMOD	chg file modes	4	CHMOD
AFC_CHOWN	chg owner and grp of a file	5	CHOWN
AFC_DUB	init a process	6	DUB
AFC_EXEC	execute a file with setid	7	EXEC
AFC_FCHAUDIT_U	chg user audit options when file is open	8	FCHAUDIT
AFC_FCHMOD	chg file modes when file is open	9	FCHMOD
AFC_FCHOWN	chg owner and group of file when open	10	FCHOWN
AFC_GETCWD	get current working directory	11	GETCWD
AFC_GETPSENT	get process entry	12	GETPSENT
AFC_KILL	signal a process	13	KILL
AFC_LINK	link to a file	14	LINK
AFC_LSTAT	get file status don't resolve ending symlink	15	LSTAT
AFC_MKDIR	make a directory	16	MKDIR
AFC_MKNOD	make a file node	18	MKNOD
AFC_MOUNT	mount a file system	18	MOUNT
AFC_OPEN	open a file	19	OPEN
AFC_OPENDIR	open a directory	20	OPENDIR

Table 191. OpenExtensions Audit Function Codes (continued)

Name	Description	Value	Character String
AFC_PATHCONF	get configurable path name variables	21	PATHCONF
AFC_PTRACE	debug a process	22	PTRACE
AFC_READLINK	read a symbolic link	23	READLINK
AFC_RENAME	rename a file	26	RENAME
AFC_RMDIR	remove a directory	25	RMDIR
AFC_SETEGID	set effective GID	26	SETEGID
AFC_SETEUID	set effective UID	27	SETEUID
AFC_SETGID	set real/saved and/or effective GID	28	SETGID
AFC_SETUID	set real/saved and/or effective UID	29	SETUID
AFC_STAT	get file status	30	STAT
AFC_SYMLINK	create a symbolic link	31	SYMLINK
AFC_UNLINK	remove directory entrs (delete a file)	32	UNLINK
AFC_UNMOUNT	unmount a file system	33	UNMOUNT
AFC_UTIME	set file access/modification times	34	UTIME
AFC_UNDUB_EXIT	terminate a process	35	UNDUB/_EXIT
AFC_WRITE	write to a file (clear setid bits)	36	WRITE
AFC_CHAUDIT_A	chg auditor audit opts	37	CHAUDIT
AFC_FCHAUDIT_A	chg auditor audit opts when file is open	38	FCHAUDIT
AFC_LOOKUP	path name resolution	39	LOOKUP
AFC_TTYNAME	get pathname of term	40	TTYNAME
AFC_IOCTL	get path name	41	IOCTL
AFC_GETMNT	get mount entry	42	
AFC_QUIESCE	quiesce mount	43	QUIESCE
AFC_UNQUIESCE	unquiesce mount	44	UNQUIESCE
AFC_VREGISTER	server registration	45	VREGISTER
AFC_VRESOLVEPN	server resolve path name	46	VRESOLVEPN
AFC_VLOOKUP	server lookup	47	VLOOKUP
AFC_VREADWRITE	server rd write	48	VREADWRITE
AFC_VREADDIR	server read directory	49	VREADDIR
AFC_SIGACTION	change Osigset action	50	SIGACTION
AFC_CREATE	server create	51	VCREATE
AFC_VMAKEDIR	server make directory	52	VMAKEDIR
AFC_VSYMLINK	server symbolic link	53	VSYMLINK
AFC_VSETATTR	server set file attributes	54	VSETATTR

Table 191. OpenExtensions Audit Function Codes (continued)

Name	Description	Value	Character String
AFC_VLINK	server link	55	VLINK
AFC_VREMOVEDIR	server remove directory	56	VREMOVEDIR
AFC_VREMOVE	server remove	57	VREMOVE
AFC_VRENAME	server rename	58	VRENAME
AFC_CHATTR	change file attributes	59	CHATTR
AFC_FCHATTR	change file attributes for open file	60	FCHATTR
AFC_THLMT	set thread limit	61	THLMT
AFC_MSGCTL	message control	62	MSGCTL
AFC_MSGGET	message obtain	63	MSGGET
AFC_MSGRCV	message receive	64	MSGRCV
AFC_MSGSND	message send	65	MSGSND
AFC_SEMCTL	semaphore control	66	SEMCTL
AFC_SEMGET	get set of semaphores	67	SEMGET
AFC_SEMOP	semaphore operations	68	SEMOP
AFC_SHMAT	shared memory attach	69	SHMAT
AFC_SHMCTL	shared memory control	70	SHMCTL
AFC_SETREGID	set real and/or effective GID	71	SETREGID
AFC_SHMGET	shared memory get	72	SHMGET
AFC_WGETIPC	query IPC status	73	W_GETIPC
AFC_REMOVE	remove	74	RPC_RMID
AFC_SET_MODE	set mode	75	IPC_SET
AFC_SET_MSGQB	set msg queue max bytes	76	IPC_SET
AFC_SET_GID	set supplementary groups	77	SETGROUPS
AFC_PASSWORD	verify password	78	_PASSWD
AFC_LCHOWN	change owner and group of a symbolic link	79	LCHOWN
AFC_TRUNCATE	truncate a file	80	TRUNCATE
AFC_PFSCTL	control function for the phys. file system	81	PFSCTL
AFC_SETRLIMIT	set maximum resource consumption	82	SETRLIMIT
AFC_SETPRIORITY	set process scheduling priority	83	SETPRIORITY
AFC_NICE	change priority of a process	84	NICE
AFC_SETREUID	set real and/or effective UID	85	SETREUID
AFC_WRITEV	write on a file	86	WRITEV
AFC_FCHDIR	change working directory	87	FCHDIR

Table 191. OpenExtensions Audit Function Codes (continued)

Name	Description	Value	Character String
AFC_CHROOT	change root directory	88	CHROOT
AFC_REALPATH	resolve path name	89	REALPATH
AFC_STATVFS	get file system information	90	STATVFS
AFC_BIND	bind a name to a socket	91	BIND
AFC_SOCKET	create an endpoint for communication	92	SOCKET
AFC_THREAD_SEC	thread level security	93	THREAD_SEC
AFC_AUTHCHECK	authority check	94	AUTHCHECK
AFC_ACC_SEND	send access rights	95	ACC_SEND
AFC_ACC_RECV	receive access rights	96	ACC_RECV
AFC_ACC_DISC	discard access rights	97	ACC_DISC
AFC_NEWGRP	newgrp shell utility	98	NEWGRP

Notices

This information was developed for products and services offered in the US. This material might be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

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 grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
US

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

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 jurisdictions 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 websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you provide 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 Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
US

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 document 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.

The performance data and client examples cited are presented for illustrative purposes only. Actual performance results may vary depending on specific configurations and operating conditions.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information may contain examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to actual people or business enterprises is entirely coincidental.

COPYRIGHT LICENSE:

This information may contain sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Programming Interface Information

This publication primarily documents intended programming interfaces that allow the customer to write programs to obtain services of an external security manager.

This document also contains information that is NOT intended to be used as programming interfaces. This information is identified where it occurs, either by an introductory statement to a chapter or section or by the following marking:

NOT programming interface information

End of NOT programming interface information

Trademarks

IBM, the IBM logo, and ibm.com® are trademarks or registered trademarks of International Business Machines Corp., in the United States and/or other countries. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on IBM Copyright and trademark information (https://www.ibm.com/legal/copytrade).

Terms and Conditions for Product Documentation

Permissions for the use of these publications are granted subject to the following terms and conditions.

Applicability

These terms and conditions are in addition to any terms of use for the IBM website.

Personal Use

You may reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative work of these publications, or any portion thereof, without the express consent of IBM.

Commercial Use

You may reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these publications, or reproduce, distribute or display these publications or any portion thereof outside your enterprise, without the express consent of IBM.

Rights

Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or, as determined by IBM, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.

IBM Online Privacy Statement

IBM Software products, including software as a service solutions, ("Software Offerings") may use cookies or other technologies to collect product usage information, to help improve the end user experience, to tailor interactions with the end user, or for other purposes. In many cases no personally identifiable information is collected by the Software Offerings. Some of our Software Offerings can help enable you to collect personally identifiable information. If this Software Offering uses cookies to collect personally identifiable information, specific information about this offering's use of cookies is set forth below.

This Software Offering does not use cookies or other technologies to collect personally identifiable information.

If the configurations deployed for this Software Offering provide you as customer the ability to collect personally identifiable information from end users via cookies and other technologies, you should seek your own legal advice about any laws applicable to such data collection, including any requirements for notice and consent.

For more information about the use of various technologies, including cookies, for these purposes, see:

- The section entitled **IBM Websites** at IBM Privacy Statement (https://www.ibm.com/privacy)
- Cookies and Similar Technologies (https://www.ibm.com/privacy#Cookies_and_Similar_Technologies)

Bibliography

This topic lists the publications in the z/VM library. For abstracts of the z/VM publications, see *z/VM: General Information*.

Where to Get z/VM Information

The current z/VM product documentation is available in IBM Documentation - z/VM (https://www.ibm.com/docs/en/zvm).

z/VM Base Library

Overview

- *z/VM: License Information*, GI13-4377
- *z/VM: General Information*, GC24-6286

Installation, Migration, and Service

- *z/VM: Installation Guide*, GC24-6292
- *z/VM: Migration Guide*, GC24-6294
- *z/VM: Service Guide*, GC24-6325
- *z/VM: VMSES/E Introduction and Reference*, GC24-6336

Planning and Administration

- *z/VM: CMS File Pool Planning, Administration, and Operation*, SC24-6261
- *z/VM: CMS Planning and Administration*, SC24-6264
- *z/VM: Connectivity*, SC24-6267
- *z/VM: CP Planning and Administration*, SC24-6271
- *z/VM: Getting Started with Linux on IBM Z*, SC24-6287
- *z/VM: Group Control System*, SC24-6289
- *z/VM: I/O Configuration*, SC24-6291
- *z/VM: Running Guest Operating Systems*, SC24-6321
- *z/VM: Saved Segments Planning and Administration*, SC24-6322
- *z/VM: Secure Configuration Guide*, SC24-6323

Customization and Tuning

- *z/VM: CP Exit Customization*, SC24-6269
- *z/VM: Performance*, SC24-6301

Operation and Use

- *z/VM: CMS Commands and Utilities Reference*, SC24-6260
- *z/VM: CMS Primer*, SC24-6265
- *z/VM: CMS User's Guide*, SC24-6266
- *z/VM: CP Commands and Utilities Reference*, SC24-6268

- *z/VM: System Operation*, SC24-6326
- *z/VM: Virtual Machine Operation*, SC24-6334
- *z/VM: XEDIT Commands and Macros Reference*, SC24-6337
- *z/VM: XEDIT User's Guide*, SC24-6338

Application Programming

- *z/VM: CMS Application Development Guide*, SC24-6256
- *z/VM: CMS Application Development Guide for Assembler*, SC24-6257
- *z/VM: CMS Application Multitasking*, SC24-6258
- *z/VM: CMS Callable Services Reference*, SC24-6259
- *z/VM: CMS Macros and Functions Reference*, SC24-6262
- *z/VM: CMS Pipelines User's Guide and Reference*, SC24-6252
- *z/VM: CP Programming Services*, SC24-6272
- *z/VM: CPI Communications User's Guide*, SC24-6273
- *z/VM: ESA/XC Principles of Operation*, SC24-6285
- *z/VM: Language Environment User's Guide*, SC24-6293
- *z/VM: OpenExtensions Advanced Application Programming Tools*, SC24-6295
- *z/VM: OpenExtensions Callable Services Reference*, SC24-6296
- *z/VM: OpenExtensions Commands Reference*, SC24-6297
- *z/VM: OpenExtensions POSIX Conformance Document*, GC24-6298
- *z/VM: OpenExtensions User's Guide*, SC24-6299
- *z/VM: Program Management Binder for CMS*, SC24-6304
- *z/VM: Reusable Server Kernel Programmer's Guide and Reference*, SC24-6313
- *z/VM: REXX/VM Reference*, SC24-6314
- *z/VM: REXX/VM User's Guide*, SC24-6315
- *z/VM: Systems Management Application Programming*, SC24-6327
- *z/VM: z/Architecture Extended Configuration (z/XC) Principles of Operation*, SC27-4940

Diagnosis

- *z/VM: CMS and REXX/VM Messages and Codes*, GC24-6255
- *z/VM: CP Messages and Codes*, GC24-6270
- *z/VM: Diagnosis Guide*, GC24-6280
- *z/VM: Dump Viewing Facility*, GC24-6284
- *z/VM: Other Components Messages and Codes*, GC24-6300
- *z/VM: VM Dump Tool*, GC24-6335

z/VM Facilities and Features

Data Facility Storage Management Subsystem for z/VM

- *z/VM: DFSMS/VM Customization*, SC24-6274
- *z/VM: DFSMS/VM Diagnosis Guide*, GC24-6275
- *z/VM: DFSMS/VM Messages and Codes*, GC24-6276
- *z/VM: DFSMS/VM Planning Guide*, SC24-6277

- *z/VM: DFSMS/VM Removable Media Services*, SC24-6278
- *z/VM: DFSMS/VM Storage Administration*, SC24-6279

Directory Maintenance Facility for z/VM

- *z/VM: Directory Maintenance Facility Commands Reference*, SC24-6281
- *z/VM: Directory Maintenance Facility Messages*, GC24-6282
- *z/VM: Directory Maintenance Facility Tailoring and Administration Guide*, SC24-6283

Open Systems Adapter

- Open Systems Adapter/Support Facility on the Hardware Management Console (https://www.ibm.com/docs/en/SSLTBW_2.3.0/pdf/SC14-7580-02.pdf), SC14-7580
- Open Systems Adapter-Express ICC 3215 Support (https://www.ibm.com/docs/en/zos/2.3.0?topic=osa-icc-3215-support), SA23-2247
- Open Systems Adapter Integrated Console Controller User's Guide (https://www.ibm.com/docs/en/SSLTBW_2.3.0/pdf/SC27-9003-02.pdf), SC27-9003
- Open Systems Adapter-Express Customer's Guide and Reference (https://www.ibm.com/docs/en/SSLTBW_2.3.0/pdf/ioa2z1f0.pdf), SA22-7935

Performance Toolkit for z/VM

- *z/VM: Performance Toolkit Guide*, SC24-6302
- *z/VM: Performance Toolkit Reference*, SC24-6303

The following publications contain sections that provide information about z/VM Performance Data Pump, which is licensed with Performance Toolkit for z/VM.

- *z/VM: Performance*, SC24-6301. See z/VM Performance Data Pump.
- *z/VM: Other Components Messages and Codes*, GC24-6300. See Data Pump Messages.

RACF Security Server for z/VM

- *z/VM: RACF Security Server Auditor's Guide*, SC24-6305
- *z/VM: RACF Security Server Command Language Reference*, SC24-6306
- *z/VM: RACF Security Server Diagnosis Guide*, GC24-6307
- *z/VM: RACF Security Server General User's Guide*, SC24-6308
- *z/VM: RACF Security Server Macros and Interfaces*, SC24-6309
- *z/VM: RACF Security Server Messages and Codes*, GC24-6310
- *z/VM: RACF Security Server Security Administrator's Guide*, SC24-6311
- *z/VM: RACF Security Server System Programmer's Guide*, SC24-6312
- *z/VM: Security Server RACROUTE Macro Reference*, SC24-6324

Remote Spooling Communications Subsystem Networking for z/VM

- *z/VM: RSCS Networking Diagnosis*, GC24-6316
- *z/VM: RSCS Networking Exit Customization*, SC24-6317
- *z/VM: RSCS Networking Messages and Codes*, GC24-6318
- *z/VM: RSCS Networking Operation and Use*, SC24-6319
- *z/VM: RSCS Networking Planning and Configuration*, SC24-6320

TCP/IP for z/VM

- *z/VM: TCP/IP Diagnosis Guide*, GC24-6328
- *z/VM: TCP/IP LDAP Administration Guide*, SC24-6329
- *z/VM: TCP/IP Messages and Codes*, GC24-6330
- *z/VM: TCP/IP Planning and Customization*, SC24-6331
- *z/VM: TCP/IP Programmer's Reference*, SC24-6332
- *z/VM: TCP/IP User's Guide*, SC24-6333

Prerequisite Products

Device Support Facilities

- Device Support Facilities (ICKDSF): User's Guide and Reference (https://www.ibm.com/docs/en/SSLTBW_2.5.0/pdf/ickug00_v2r5.pdf), GC35-0033

Related Products

XL C++ for z/VM

- *XL C/C++ for z/VM: Runtime Library Reference*, SC09-7624
- *XL C/C++ for z/VM: User's Guide*, SC09-7625

z/OS

IBM Documentation - z/OS (https://www.ibm.com/docs/en/zos)

Index

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

Y

Z

Product Number: 5741-A09

Printed in USA

SC24-6309-74