z/OS
2.5

Security Server RACF Diagnosis Guide

GA32-0886-50

Note

Before using this information and the product it supports, read the information in “Notices” on page 273.

This edition applies to Version 2 Release 5 of z/OS® (5650-ZOS) and to all subsequent releases and modifications until otherwise indicated in new editions.

Last updated: 2023-06-28

Contents

Figures

Tables

About this document

This document supports z/OS (5650-ZOS) and contains information about Resource Access Control Facility (RACF®), which is part of z/OS Security Server.

This document provides information that is useful for diagnosing problems in RACF, such as:

- Parameter list descriptions
- Process flows
- Control block overviews
- SVC dump title descriptions
- A description of variable recording area (VRA) keys provided by RACF

The information included in this document should help you to:

- Follow diagnostic procedures for each type of problem in RACF
- Collect and analyze data needed to diagnose the problem
- Develop a search argument to be used for searching problem-reporting databases
- Gather the necessary problem data before reporting the problem to IBM®

In addition, this document describes how to use the BLKUPD command to correct problems in the RACF database.

Intended audience

This document is for anyone who diagnoses problems that appear to be caused by RACF and for RACF system programmers who intend to use the BLKUPD command to correct problems in the RACF database.

This document assumes that you:

- Understand basic system concepts and the use of system services
- Code in assembler language and read assembler and linkage editor output
- Understand the commonly used diagnostic tasks and aids, such as message logs, system dumps, and the interactive problem control system (IPCS)
- Understand the externals for RACF

How to use this document

Before using this document, collect this problem data:

- The problem type, such as an abend
- An indication that the problem was caused by RACF

If you do not have this data, see the problem determination and diagnosis publications for your system and perform the recommended procedures.

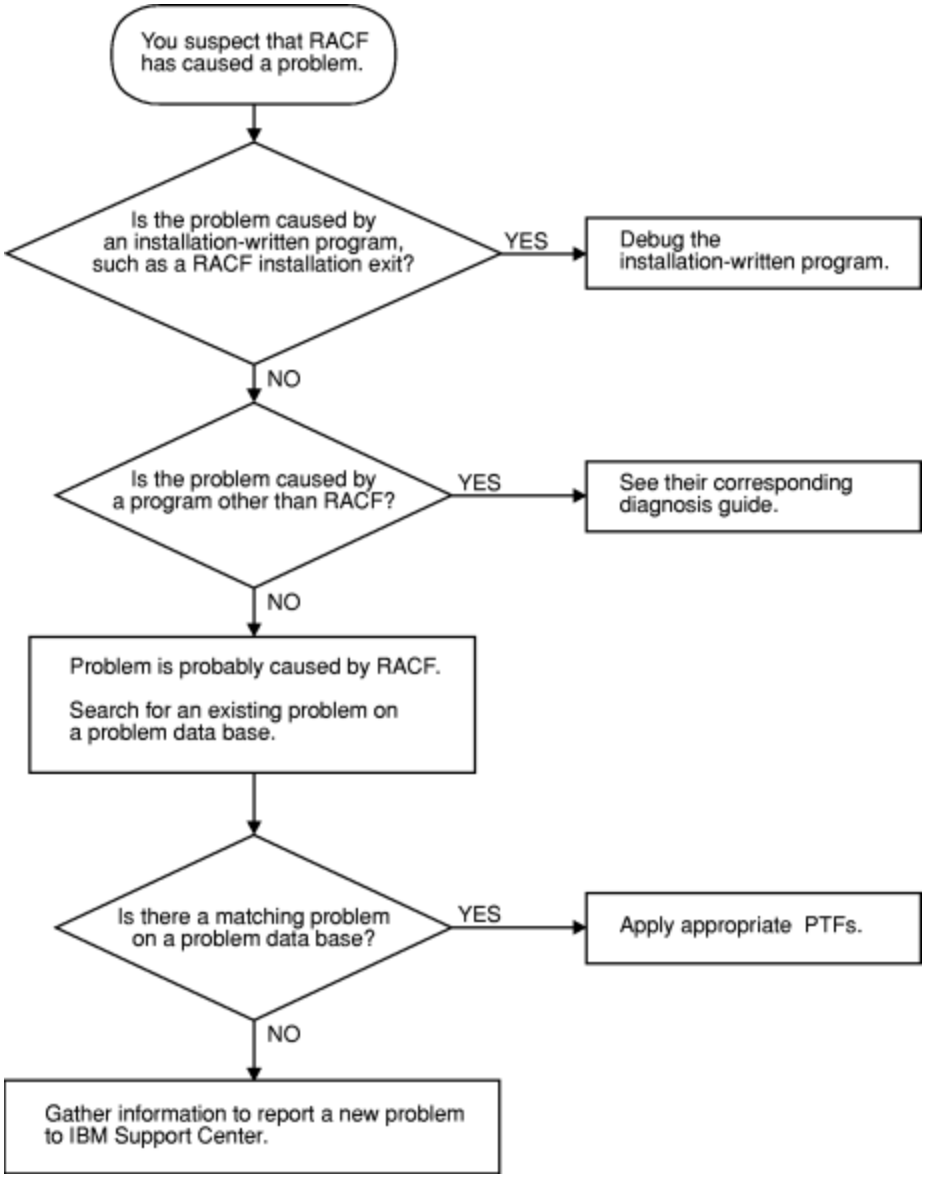
Use this document to diagnose problems in RACF only. If the problem is not caused by RACF, return to your system diagnosis guide to identify the failing component or program.

Use this document to diagnose problems in RACF as follows:

1. Identify the problem type.
2. Collect problem data.
3. Analyze the problem data to develop symptoms.
4. Develop search arguments, search problem-reporting databases, and request the problem fix if the problem has been reported before. If not, continue diagnosis.

5. Collect additional problem data.
6. Analyze the problem data to isolate the problem.
7. Report the problem to IBM if assistance is needed or if the problem is new.

This flowchart illustrates the possible paths to be taken during problem analysis while using this document.



Where to find RACF information

For general RACF information, see the RACF home page (www.ibm.com/products/resource-access-control-facility/resources).

For more detailed RACF information, see *z/OS Security Server RACF System Programmer's Guide*. For help in gathering information about your RACF installation, see your RACF security administrator and RACF auditor or see *z/OS Security Server RACF Security Administrator's Guide* and *z/OS Security Server RACF Auditor's Guide*.

For the syntax, return codes, and error reason codes for RACF macros, see *z/OS Security Server RACF Macros and Interfaces* except for the RACROUTE macro, which is described in *z/OS Security Server RACROUTE Macro Reference*.

For the mappings of RACF control blocks (such as the RCVT and the ACEE), see *z/OS Security Server RACF Data Areas* in the z/OS Internet library (www.ibm.com/servers/resourcelink/svc00100.nsf/pages/zosInternetLibrary).

Where to find more information

When possible, this information uses cross-document links that go directly to the topic in reference using shortened versions of the document title. For complete titles and order numbers of the documents for all products that are part of z/OS, see *z/OS Information Roadmap*.

To find the complete z/OS library, including the z/OS Documentation, see the z/OS Internet library (www.ibm.com/servers/resourcelink/svc00100.nsf/pages/zosInternetLibrary).

To find educational material, see the IBM Education home page (www.ibm.com/training).

RACF courses

The following RACF classroom courses are available in the United States:

ES191
Basics of z/OS RACF Administration

BE870
Effective RACF Administration

ES885
Exploiting the Advanced Features of RACF

IBM provides various educational offerings for RACF. For more information about classroom courses and other offerings, do any of the following:

- See your IBM representative
- Call 1-800-IBM-TEACH (1-800-426-8322)

Other sources of information

IBM provides customer-accessible discussion areas where RACF may be discussed by customer and IBM participants. Other information is also available through the Internet.

Internet sources

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

- z/OS Internet library (www.ibm.com/servers/resourcelink/svc00100.nsf/pages/zosInternetLibrary)
- IBM Redbooks (www.ibm.com/redbooks)
- Enterprise security (www.ibm.com/systems/z/solutions/enterprise-security.html)
- RACF home page (www.ibm.com/products/resource-access-control-facility/resources)
- RACF download page (github.com/IBM/IBM-Z-zOS/tree/master/zOS-RACF/Downloads)

How to send your comments to IBM

We invite you to submit comments about the z/OS product documentation. Your valuable feedback helps to ensure accurate and high-quality information.

Important: If your comment regards a technical question or problem, see instead "If you have a technical problem" on page xvii.

Submit your feedback by using the appropriate method for your type of comment or question:

Feedback on z/OS function
If your comment or question is about z/OS itself, submit a request through the IBM RFE Community (www.ibm.com/developerworks/rfe/).

Feedback on IBM Documentation
If your comment or question is about the IBM Documentation functionality, for example search capabilities or how to arrange the browser view, send a detailed email to IBM Documentation at ibmdoc@us.ibm.com.

Feedback on the z/OS product documentation and content
If your comment is about the information that is provided in the z/OS product documentation library, send a detailed email to mhvrcfs@us.ibm.com. We welcome any feedback that you have, including comments on the clarity, accuracy, or completeness of the information.

To help us better process your submission, include the following information:

- Your name, company/university/institution name, and email address
- The following deliverable title and order number: z/OS Security Server RACF Diagnosis Guide, GA32-0886-50
- The section title of the specific information to which your comment relates
- The text of your comment.

When you send comments to IBM, you grant IBM a nonexclusive authority to use or distribute the comments in any way appropriate without incurring any obligation to you.

IBM or any other organizations use the personal information that you supply to contact you only about the issues that you submit.

If you have a technical problem

If you have a technical problem or question, do not use the feedback methods that are provided for sending documentation comments. Instead, take one or more of the following actions:

- Go to the IBM Support Portal (support.ibm.com).
- Contact your IBM service representative.
- Call IBM technical support.

Summary of changes

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.

Note: IBM z/OS policy for the integration of service information into the z/OS product documentation library is documented on the z/OS Internet Library under IBM z/OS Product Documentation Update Policy (www-01.ibm.com/servers/resourcelink/svc00100.nsf/pages/ibm-zos-doc-update-policy?OpenDocument).

Summary of changes for z/OS Version 2 Release 5 (V2R5)

The following changes are made to z/OS Version 2 Release 5 (V2R5).

New

The following information is new.

February 2023

- To indicate whether the LOGSTRX option was specified for a RACROUTE REQUEST=FASTAUTH invocation, the LOGSTRX bit is added to the “FASTAUTH service parameter list” on page 168.

V2R5

- Added RACF release level V2R5 and RACF FMID HRF77D0 to the 'RACF FMIDs' table. See “Examples of search arguments for RACF” on page 44.
- Added new constant HRF77D0. See “Constants” on page 253.
- Added information on diagnosing a problem that is associated with a RACF VSAM database. See: Chapter 4, “Troubleshooting your RACF database,” on page 105
- ICBEXTND description was updated. See Appendix A, “The Inventory Control Block (ICB),” on page 243.

Changed

No information is changed in this release.

Deleted

No information is deleted in this release.

Summary of changes for z/OS Version 2 Release 4 (V2R4)

The following changes are made to z/OS Version 2 Release 4 (V2R4).

New

The following information is new.

Prior to June 2020 refresh

- The segment identifiers table has been updated to add X'04-CSDATA to the data set column and X'12'-MFA, X'13'-MFPOLICY, X'14-CSDATA, X'15'-IDTPARMS, and X'16'-JES to the general column. For more information, see“Index entries” on page 111.

Changed

No information has been changed in this edition.

Deleted

No information has been deleted in this edition.

Summary of changes for z/OS Version 2 Release 3 (V2R3)

The following changes are made to z/OS Version 2 Release 3 (V2R3).

New

There is no new information in this version.

Changed

- “Trace example 1” on page 50 and “Trace example 2” on page 52 have been updated to include PTFs UA83891 (7A0), UA83892 (790), UA83893 (780)

Chapter 1. Starting diagnosis

This information contains diagnostic procedures that you can perform with problem data that is quick and easy to obtain. It includes:

Planning for problem diagnosis

Consider making these preparations for diagnosis before you use RACF:

- Install and operate RACF so that you get adequate problem data (such as messages and dumps) when problems occur.
- Provide *timely* and *complete* backups of the RACF database.
- Obtain access to a RACF user with the SPECIAL attribute.
- Reserve a RACF user ID with the SPECIAL attribute for use **only after logon problems are resolved**. For example, if all users logging on are, through an error, revoked when logging on, then even the system security administrator could be revoked when logging on. After the problem is corrected, the system security administrator could then log on with the user ID that is still active and activate the other user IDs.
- Obtain messages that have been sent to the system console or the security console.
- Prevent common problems by using RACF macros and utilities. See the recommendations in "Utilities" on page 3.
- Correct any problems that were caused while using RACF profiles and options. See *z/OS Security Server RACF Security Administrator's Guide*.

Troubleshooting the problem

This describes the RACF information you will need to troubleshoot your problem.

- **"Diagnosis hints" on page 2.**

  This information includes:

  - How to make sure that the problem is in RACF
  - How to prevent some common programming errors in RACF macros and utilities

- **"Troubleshooting tables for problems in RACF" on page 7.**

  These tables provide steps to follow in diagnosing each type of symptom, such as abend, message, or incorrect output.

- **"Performing a preliminary search for a matching problem" on page 41.**

  This information tells you how to search a database for a matching problem.

  Note:

  Chapter 4, "Troubleshooting your RACF database," on page 105 describes how to use the BLKUPD command. The skills used in working with the BLKUPD command are different from the skills used in general troubleshooting for suspected RACF problems.

Diagnosis hints

This information provides advice on making sure that this is a problem with RACF and describes special types of problems that you might be able to fix without calling IBM.

If the recommendations that follow do not solve the problem, you need to follow the other diagnosis procedures described in the troubleshooting tables.

Making sure this is a RACF problem

Table 1 on page 2 helps you make sure that the problem is in RACF, rather than in either the caller of RACF or a system service used during processing of a RACF request.

For some problems, you might be able to do a quick check of the problem symptoms in the dump (if present), the job and system message log, SYS1.LOGREC, trace records, or other problem output.

Table 1. Determining whether you have a RACF problem

Questions	Recommendations
Was there a *valid* request for a RACF function?	This could be that: • A RACF macro was invoked • A RACF utility was invoked • A RACF command was invoked • A user selected a RACF option on an ISPF panel The RACF request *must be valid*. A user request could invoke a routine that in turn invokes RACF. A RACF error message (like ICH408I, insufficient access authority) would then be issued not because of a RACF error or because of an error made by the original user, but because of the routine that invoked RACF. **Note:** If there is a system dump, you can check this in the trace records that are recorded in the dump. If there is no system dump, you may need to re-create the problem and request a system dump when the problem recurs.
Were error messages issued by system services that were used during RACF processing?	Check whether any error messages were issued for the user request, and make sure that message IDs are included with the messages. If a batch job produced the output, check that the job statement had MSGLEVEL=(1,1) specified. If a TSO/E user had the problem, check that the user profile had these options specified: PROFILE WTPMSG MSGID.
Did RACF return control to the program that made the request?	For RACF macros and utilities: if there is a system dump, you can check this in the trace records that are recorded in the dump or (if an SVC was issued) in the RB chain in the system dump. If there is no system dump, you might need to re-create the problem and request a system dump when the problem recurs. For batch jobs, you can also check whether the job step for the program completed with a zero return code. (If an application program made the request, the TSO TEST command, a SLIP command issued by the system operator, or a WTO or PTRACE macro added to the program could be helpful in checking the execution steps within the program.)

Note: While diagnosing the problem, you need to continue to check whether something other than RACF might be the source of the problem. Possible candidates are:

- Callers of RACF
- System services used by RACF

- A hardware error
- A program that overlaid storage used by RACF, its caller, or a system service
- A storage shortage can cause RACF commands to ABEND unpredictably

Common usage problems with RACF macros and utilities

This information provides advice for resolving problems that might result when using the RACF macros and utilities. The problems described here might appear at first to be errors in RACF.

Note: Advice for resolving coding problems is in *z/OS Security Server RACF System Programmer's Guide*.

Utilities

This information describes typical errors for RACF utilities and advice on action to take to resolve the problem.

IRRUT100 Utility

Error
Assuming that the utility is in a loop because it runs a long time. This utility reads every profile in the RACF database and usually takes a long time to run.

Action
Run the utility when the system is not heavily used and allow it to continue running until completion.

IRRUT200 Utility

Error
Assuming the RACF database contains no errors because the utility runs without error. The utility does not find all errors on a RACF database. Errors might exist within profiles and could result in message IRR407I.

```
RACF DATA SET INDEX ERROR. xxx
RACF DATA SET SEQUENCE nnn, dsname SEARCH ON entry-name
```

where *xxx* is PRIMARY or BACKUP, *nnn* is the database sequence number, *dsname* is the data set name, and *entry-name* is the type of entry.

Action
If the utility does not find any errors, list the profiles involved using the appropriate RACF command. Look for error messages such as NO CONNECT ENTRY FOUND or incorrect data in fields such as OWNER.

Use the RACF commands to list and correct the profiles, if possible. If the commands fail, use BLKUPD to correct or delete the profiles.

RACF Sysplex Data Sharing: If RACF is enabled for sysplex communication and the system is operating in read-only mode, you cannot use BLKUPD to update the RACF database.

You can, however, run BLKUPD from another system (in the same data sharing group) that is not in read-only mode.

Error
Running the utility against an active output database. Even if there are no other users on the system, other system tasks and applications might be accessing the RACF database. If any changes are made to the database, the output of the utility is not reliable.

Action
Enter the RVARY INACTIVE or RVARY SWITCH command before running the utility.

Error
Assuming that the run of the utility failed because a nonzero return code was received.

Action
The return code returned is the highest return code encountered while processing the RACF database. A nonzero return code does not mean that the utility failed. See *z/OS Security Server RACF Messages and Codes* for details.

IRRUT400 Utility

Error
Do not use this utility to merge RACF databases from separate systems.

Action
Use it only for merging back together a database from one system.

Error
Ignoring return code 4 or return code 8 on IRRUT400 MERGE. These return codes can indicate that duplicate profiles were found. Ignoring them can result in profile conflicts.

Action
Check the duplicate profiles on each of the input databases to determine which ones you want on the output database. Use the RACF commands to delete the unwanted ones and rerun the IRRUT400 MERGE.

Error
Running the utility against an active RACF database. Even if there are no other users on the system, other system tasks and applications might be accessing the RACF database. If any changes are made to the database, the output of the utility is not reliable.

Action
Enter the RVARY INACTIVE or RVARY SWITCH command before running the utility.

Error
RACF manager return code X'50' (decimal 80) received while trying to access the RACF database. This return code means that one of these occurred:

- You attempted to update a RACF database that has had the utility run against it with the LOCKINPUT keyword specified.
- RACF is enabled for sysplex communication and while running in read-only mode, you tried to update the RACF database.

Action
The LOCKINPUT keyword makes the input database unusable for system updates. It should be used with caution.

If your RACF database is locked, and you want to unlock it, use the UNLOCKINPUT parameter of IRRUT400 or IRRDBU00.

Note: If this error occurs because your RACF sysplex data sharing system is in read-only mode, UNLOCKINPUT does not correct the problem. You cannot update the RACF database when your system is operating in read-only mode.

IRRDBU00 Utility

Error
RACF manager return code X'50' (decimal 80) received from another task while trying to access the RACF database. This return code means that one of these has occurred:

- A task attempted to update the RACF database while the IRRDBU00 utility was running against the database with the LOCKINPUT keyword specified.
- RACF is enabled for sysplex communication and was operating in read-only mode, when a task tried to update the RACF database.

Action
A task cannot update the database that is shared by a sysplex data sharing group when the system is operating in read-only mode. If, however, the utility is running with LOCKINPUT specified, do:

- Wait until the utility is finished running.
- Run the utility at another time when updates are *not* needed.
- Run the utility against a copy of the RACF database—not against the active database.

Note: The LOCKINPUT keyword makes the input database for the utility temporarily unusable for system updates. It should be used with caution.

Error

I/O error while reading the database; your data set has slipped into secondary extents. These console messages appear:

```
13.31.20 JOB00019 *IRR401I 500 ABEND DURING RACF PROCESSING OF
                   NEXT REQUEST FOR ENTRY MURPHYB
13.31.20 JOB00019  IRR401I 500 ABEND DURING RACF PROCESSING
```

These messages appear from SYSPRINT:

```
IRR67122I *** Utility ESTAE error routine in control. ***
IRR67124I Profile processing DID NOT finish before the abend.
          Output is NOT complete.
IRR67120I 500 abend during utility processing. Reason code 0000.
IRR67125I Utility ESTAE error routine will now attempt clean-up
          processing.
IRR67121I The module in control at time of abend was IRRDBU03.
```

Action

Reallocate the database that is causing the problem as a single extent. Recopy the data.

Error

Not all of your data is unloaded. Entire classes are missing.

Action

If you have imported a database from another system, you must use the class descriptor table from that system. IPL your system with the class descriptor table associated with the imported database.

Error

Fields are missing or field offsets are incorrect in the output database.

Action

Ensure that you have the current level of templates for the input database. Run IRRMIN00 with PARM=UPDATE and the current level of templates.

IRRRID00 Utility

Error

The RACF remove ID utility (IRRRID00) does not issue any user abends. If a user abend is received, it was issued by a called service such as DFSORT.

Action

If you see a user abend, consult the documentation for the called service.

RACF report writer utility (RACFRW)

Error

The RACF report writer utility (RACFRW) does not issue any user abends. If a user abend is received, it was issued by a called service such as SORT.

Action

If you see a user abend, consult the documentation for the called service.

IRRIRA00

Error

If the utility runs for a long period of time it is considered to be "in a loop". When going from stage 0 to stage 1 (which can take a considerable amount of time), IRRIRA00 reads every mapping profile for the application identity mapping fields in the RACF database. It then reads the base profile named by the mapping profile to verify that the base profile-alias correlation is valid, and also to verify every

user and group profile with an OMVS segment. For every valid correlation found, the alias index entry must be maintained. Additionally, when going from stage 2 to stage 3 (which can take a considerable amount of time), all application identity mapping profiles in the RACF database are deleted.

Action
Run the utility when the system is not being heavily used, and allow it to continue running through completion.

RACF macros

This information describes typical errors for RACF macros and advice on action to take to resolve the problem.

General

Error
Unexpected return code or incorrect output when using the list and execute form of a macro. Results are not the same as for standard form.

Action

1. Check that the list form is in a CSECT, not a DSECT. If the list form is only in a DSECT, the parameter list is not initialized.
2. Check that the list form was copied over correctly into the GETMAIN work area.
3. Make sure the LIST and EXECUTE forms of the RACROUTE macro specify the same value for the RELEASE= parameter.

Error
Incorrect output from a macro with some keywords that were ignored.

Action
If the macro is continued on a second line, check that there is a continuation marked in column 72 and that the second line begins in column 16. If the macro is continued in column 17 or later, the assembler ignores that line.

ICHEINTY macro

Error
Incorrect output on an ICHEINTY macro when looking at discrete or generic profiles.

Action
Check the meaning of the GENERIC keyword. This keyword refers to the encoding of the resource name. It does not correspond to the type of profile returned. If the generic profile was retrieved, the high-order bit is on at offset X'A' in the work area specified in the ICHEINTY macro. It will be off if a discrete profile was returned.

Error
Difficulties while using the ICHEINTY macro.

Action
Read the documentation of the macro very carefully. The ICHEINTY macro and its related macros are quite complex. They bypass exits, naming conventions, authorization checking, and do little parameter validation. They are recommended only when there is no other way of accomplishing the task that you want (no command or utility that does the same thing). Note that in some cases, the RACXTRT macro can be used instead of the ICHEINTY macro.

Error
Message IRR401I reporting a DEQ abend (such as abend 730) or abend 0C4 in IRRMRT00 on an ICHEINTY macro.

Action
Check the specification of ENTRY or ENTRYX. This should point to a 1-byte length followed by the ENTRY or ENTRYX NAME. If the length is zero, or if it points directly to the ENTRY or ENTRYX name, an overlay occurs, resulting in the abend.

RACROUTE macro

Error

Abend 282-64 (decimal 100) on RACROUTE REQUEST=AUTH macro with ACEE (accessor environment element) keyword, or abend 283-4C (decimal 76) on RACROUTE REQUEST=VERIFY DELETE with ACEE keyword.

Action

Check the specification of the ACEE keyword. On a REQUEST=VERIFY, the ACEE keyword specifies a pointer to the address of the ACEE. On a REQUEST=AUTH, the ACEE keyword specifies a pointer to the ACEE.

Error

Return code 4, abend 282-20 (DECIMAL 32), or abend 0C4 during parameter validation on RACROUTE REQUEST=AUTH macro.

Action

Check the specification of the ENTITY keyword. If a variable name is specified, it requires one set of parentheses. If a register is used, it requires two sets of parentheses, as in ENTITY=((R4)).

Troubleshooting tables for problems in RACF

You need to be familiar with RACF's basic concepts and functions in order to use the troubleshooting tables. If you need more information, see page “Where to find RACF information” on page xiv.

The troubleshooting tables that follow can help you:

- Select the type of problem that most closely matches your problem.
- Follow procedures to diagnose each type of problem.
- Determine the symptoms to use to search for a matching problem.
- Collect the information you need if you report the problem to IBM.

Using the troubleshooting tables

Before you begin:

You need to document the problem that you have and how it presents itself

Perform these steps to use the tables:

1. First, determine which type of symptom best describes the problem that you are troubleshooting and locate that type in the first column, **Symptom Type**, in Table 2 on page 8.

   _____

2. If you find a matching symptom type:

   a. Go to the topic listed in the second column, **Refer To Topic...**, in Table 2 on page 8. This is the appropriate troubleshooting table.

   b. In the appropriate table, locate the specific RACF symptom for the problem (for example, Abends). In the second column, **Recommended Diagnostic Procedures**, in Table 3 on page 10, follow the steps for diagnosing the problem.

   _____

3. If the problem has several symptoms, you should look up each symptom in the table. For example, some problems cause an abend and an I/O error message.

   _____

Table 2. Index to troubleshooting tables for RACF	
Symptom Type	**Refer To Topic...**
Abend	“Abend troubleshooting table” on page 9
Documentation error	“Documentation troubleshooting table” on page 23
Incorrect output	“Incorrect output troubleshooting table” on page 23
Logon problems	“Logon problems troubleshooting table” on page 33
Loop	“Loops troubleshooting table” on page 35
Messages	“Messages troubleshooting table” on page 36
Performance degradation	“Performance problems troubleshooting table” on page 37
Return code from RACF	“RACF return codes troubleshooting table” on page 39
Wait or hang	“Waits troubleshooting table” on page 40

If none of the symptoms in the list matches your problem, try these diagnostic procedures:

1. Check whether any error messages were issued for the user request and make sure that message IDs are included with the messages.

   If a batch job produced the output, check that the job statement had MSGLEVEL=(1,1) specified. If a TSO/E user had the problem, check that the user profile had these options: PROFILE WTPMSG MSGID

   _____

2. If you find an error message that might relate to the problem, see Table 8 on page 36.

   _____

3. If you do not have a system dump for the problem, and you want to request one, see “Obtaining a system dump” on page 47.

   _____

4. If you have a system dump for the problem, display or print the symptom record in the dump.

   Use the IPCS VERBEXIT LOGDATA and VERBEXIT SYMPTOMS subcommands.

   Note: SYMPTOMS has some prerequisite VERBEXITs, such as DAEDATA, which provides the abend and non-abend symptoms for the dump. You can also use the BLSCSCAN CLIST with IPCS to obtain the prerequisites. For details, see *z/OS MVS IPCS User's Guide*.

   If the symptom record indicates an abend, loop, or wait, use the corresponding entry in the list of symptom types in Table 2 on page 8.

   _____

5. If you have SYS1.LOGREC output for the problem, or a dump with LOGDATA output, check the LOGREC records related to the problem. If you have a software LOGREC record for an abend, use Table 3 on page 10.

   _____

You know that you are done when you matched a system type with the type of problem you are having.

If you still do not have a symptom type,

1. Check the general problem analysis advice in *z/OS MVS Diagnosis: Reference*.
2. Read the information in “Making sure this is a RACF problem” on page 2.
3. If you need additional assistance, report the problem to IBM.

Troubleshooting tables

Abend troubleshooting table

Table 3 on page 10 describes abends (abnormal end of task) related to RACF, including:

- **RACF abends,** which are abends that result from using RACF functions and that could occur under any operating system
- **MVS™ abends,** which are abends that MVS system services issue and that are unique to MVS

For RACF abends, the first digit indicates the type of problem, as shown in the table, and the second and third digits together indicate which SVC (supervisor call instruction) was associated with the abend. SVCs occur because a macro was invoked.

- SVC X'82' (in decimal, SVC 130) occurs because a RACHECK macro was invoked independently or by using a RACROUTE REQUEST=AUTH.
- SVC X'83' (in decimal, SVC 131) occurs because a RACINIT macro was invoked independently or by using a RACROUTE REQUEST=VERIFY.
- SVC X'84' (in decimal, SVC 132) occurs because a RACLIST macro was invoked independently or by using a RACROUTE REQUEST=LIST.
- SVC X'85' (in decimal, SVC 133) occurs because a RACDEF macro was invoked independently or by using a RACROUTE REQUEST=DEFINE.

Table 3. Troubleshooting table: Abends

Symptom	Recommended procedures for Abends
Abends: X'0Cx'	`Description:` Program exception. `Action:` 1. Format the system dump. 2. In the PSW (program status word) in the dump, find the address of the failing instruction (the "next sequential instruction"). 3. The address of the failing instruction should be an address that was available in the dump. Look in the dump for the eye-catcher before the failing instruction. (The hexadecimal values in the dump are usually interpreted along the right side of the dump. In a system dump that contains mostly unprintable characters, a readable name, such as the name of an object module, is easy to see.) The next eye-catcher before a failed instruction should include the name of the object module containing the failing instruction. 4. If the module name begins with ICH or IRR, RACF was executing when the abend occurred. Check these special cases first: **For 0C1 abends:** Use AMBLIST LISTLOAD to verify that the failing load module has been properly link-edited. • Check the JCLIN as shipped with the RACF product. • Pay particular attention to aliases and entry points. • If IGC0013{ is the failing module, it might not have been properly link-edited. **Note:** The last character in the load module name is X'C0'. Determine whether the load module has been physically moved since it was link-edited. If the load module is in SYS1.LPALIB: a. Link edit the load module again. b. Re-IPL with CLPA. **For 0C4 abends:** If the failing module is ICHCLD00, ICHDSM35, IRRRCK00, IRRRIN04, IRRRIN05, IRRRXT00, or IRRFLC00, it is likely that the module is trying to address a CSA or private profile that does not exist. This can happen when a RACROUTE REQUEST=AUTH preprocessing exit passes a return code of 4 or 8 but does not build a CSA or private profile even though the CSA or private option was specified on the AUTH request. To correct this error, correct the RACROUTE REQUEST=AUTH preprocessing exit so that it builds a CSA or private profile when requested on the AUTH request. 5. If these preliminary checks fail, check “Identifying the caller of RACF” on page 79 to see if RACF is the probable cause of the problem or to determine who called the RACF module. **Note:** To check the parameters that are actually passed to RACF on a macro request, see “Checking the parameter list passed to RACF” on page 81. 6. If your investigation leads you to think that RACF was the cause of the problem, fill out the Information to gather before calling IBM with abend problems worksheet in Table 20 on page 97 and call the IBM support center.

Table 3. Troubleshooting table: Abends (continued)	
Symptom	**Recommended procedures for Abends**
Abends: X'182' X'183' X'185'	`Description:` RACF was unable to establish an ESTAE for the SVC (supervisor call instruction). `Action:` 1. See the message issued with the abend. This could be message IRR401I or ICH409I. **Find the abend reason code, if any, in the text for the messages.** 2. See *z/OS Security Server RACF Messages and Codes* for information about the abend code and the accompanying message (if you have not looked up the explanation already). In general, the information in *z/OS Security Server RACF Messages and Codes* should be sufficient to correct the problem. 3. If the information in *z/OS Security Server RACF Messages and Codes* is not sufficient to correct the problem, do the rest of the steps in this procedure. 4. See "Performing a preliminary search for a matching problem" on page 41. If you do not find a matching problem, you need to collect problem information to report to IBM, using the steps in the rest of this list. 5. If the abend did not generate a system dump, and the problem is reproducible, obtain a system dump. See "Obtaining a system dump" on page 47. 6. Format the system dump. See "Formatting a system dump" on page 47. 7. You may want to re-create the problem you are investigating (or set up the conditions to allow the problem to recur) after setting the trap to obtain the dump. 8. See "Searching for a matching problem" on page 82. 9. If you still do not find a matching problem, see Chapter 3, "Reporting RACF problems to IBM," on page 97.
Abends: X'282' X'283' X'285'	`Description:` Parameter error on input to the SVC (supervisor call instruction). The reason codes associated with these abends indicate which parameter is bad in the parameter list. The bad parameter was probably set by the caller of RACF. `Action:` 1. See the message issued with the abend. This could be message IRR401I or ICH409I. **Find the abend reason code, if any, in the text for the messages.** 2. If you have not looked up the explanations already, see *z/OS Security Server RACF Messages and Codes* for the explanations of the message and the abend code. (Abend codes are also called "completion codes.") **Pay particular attention to the reason code associated with the abend.** In general, the information in *z/OS Security Server RACF Messages and Codes* should be sufficient to correct the problem. 3. If the information in *z/OS Security Server RACF Messages and Codes* is not sufficient to correct the problem, follow the rest of the steps in this procedure. 4. Check whether any other error messages were issued with the abend messages. These messages might indicate the context in which the abend occurred. In particular, messages preceding the abend message might indicate conditions that caused the abend. **Note:** If message IDs were not included with the messages and you can re-create the error, do: • If a batch job produced the output, check that the job statement had MSGLEVEL=(1,1) specified. • If a TSO/E user had the problem, check that the user profile had these options: PROFILE WTPMSG MSGID.

Table 3. Troubleshooting table: Abends (continued)	
Symptom	**Recommended procedures for Abends**
Abends: X'282' X'283' X'285' (continued)	1. Find the name of the **calling** module. This is the module that issued the RACF request (such as a RACF macro, command, or utility). The conditions set up by this module might have caused the abend. See "Identifying the caller of RACF" on page 79. 2. Find the owner of the module: an IBM product, a non-IBM product, or a user at your installation. If the module name begins with ICH or IRR, it is a RACF module. Continue with this procedure. If the module name begins with the letters A through I, S, or X (but not ICH or IRR), it is probably part of an IBM product, but it is not RACF. See the diagnosis or logic document for the program of which the module is a part. To determine the name of the component that owns the module, use the component summary in *z/OS MVS Diagnosis: Reference*. 3. See "Performing a preliminary search for a matching problem" on page 41. If you find a matching problem, apply the appropriate PTFs (program temporary fixes). If you do not find a matching problem, you need to collect problem information to report to IBM, using the steps in the rest of this procedure. **Please be sure to fill out a blank copy of the worksheet shown in Table 20 on page 97.** 4. If the abend did not generate a system dump, and the problem is reproducible, obtain a system dump. You may want to re-create the problem you are investigating (or set up the conditions to allow the problem to recur) after setting the trap to obtain the dump. 5. See "Searching for a matching problem" on page 82. 6. If you still do not find a matching problem, see Chapter 3, "Reporting RACF problems to IBM," on page 97.
Abends: X'382' X'383' X'385'	`Description:` A user exit associated with an SVC (supervisor call instruction) passed a non-valid return code. `Action:` 1. See the message issued with the abend. This could be message IRR401I or ICH409I. 2. If you have not looked up the explanations already, see *z/OS Security Server RACF Messages and Codes* for the explanations of the message and the abend code. (Abend codes are also called "completion codes.") **Pay particular attention to the explanation of the abend code.** In general, the information in *z/OS Security Server RACF Messages and Codes* should be sufficient to correct the problem. 3. If the information in *z/OS Security Server RACF Messages and Codes* is not sufficient to correct the problem, do the rest of the steps in this procedure. 4. Check whether any other error messages were issued with the abend messages. These messages might indicate the context in which the abend occurred. In particular, messages preceding the abend message might indicate conditions that caused the abend. **Note:** If message IDs were not included with the messages and you can re-create the error, do: • If a batch job produced the output, check that the job statement had MSGLEVEL=(1,1) specified. • If a TSO/E user had the problem, check that the user profile had these options: PROFILE WTPMSG MSGID.

Table 3. Troubleshooting table: Abends (continued)

Symptom	Recommended procedures for Abends
Abends: X'382' X'383' X'385' (continued)	1. Find the name of the **calling** module. This is the module that issued the RACF request (such as a RACF macro, command, or utility). The conditions set up by this module might have caused the abend. See "Identifying the caller of RACF" on page 79. 2. Find the owner of the module: an IBM product, a non-IBM product, or a user at your installation. If the module name begins with ICH or IRR, it is a RACF module. Continue with this procedure. If the module name begins with the letters A through I, S, or X (but not ICH or IRR), it is probably part of an IBM product, but it is not RACF. See the diagnosis or logic document for the program of which the module is a part. To determine the name of the component that owns the module, use the component summary in *z/OS MVS Diagnosis: Reference*. 3. See "Performing a preliminary search for a matching problem" on page 41. If you find a matching problem, apply the appropriate PTFs (program temporary fixes). If you do not find a matching problem, you need to collect problem information to report to IBM, using the steps in the rest of this procedure. **Please be sure to fill out a blank copy of the worksheet shown in Table 20 on page 97.** 4. If the abend did not generate a system dump, and the problem is reproducible, obtain a system dump. See "Obtaining a system dump" on page 47. 5. Format the system dump. See "Formatting a system dump" on page 47. 6. You might want to re-create the problem you are investigating (or set up the conditions to allow the problem to recur) after setting the trap to obtain the dump. 7. See "Searching for a matching problem" on page 82. 8. If you still do not find a matching problem, see Chapter 3, "Reporting RACF problems to IBM," on page 97.
Abends: X'3C7'	`Description:` RACF storage manager error during processing of a non-SVC, SRB mode, get space and free space request. `Action:` 1. See the message issued with the abend. This could be message ICH409I. **Find the abend reason code, if any, in the text for the message.** 2. If you have not looked up the explanations already, see *z/OS Security Server RACF Messages and Codes* for the explanations of the message and the abend code. (Abend codes are also called "completion codes.") **Pay particular attention to the reason code associated with the abend.** In general, the information in *z/OS Security Server RACF Messages and Codes* should be sufficient to correct the problem. 3. See Chapter 3, "Reporting RACF problems to IBM," on page 97.

Table 3. Troubleshooting table: Abends (continued)	
Symptom	**Recommended procedures for Abends**
Abends: X'400®'	`Description:` This abend is issued by IOS (I/O supervisor) when the RACF database goes into multiple extents. (The RACF database must be allocated in a single contiguous extent.) `Action:` 1. To correct this problem, reallocate the RACF database with enough space to fit in a single extent and re-IPL the system. 2. If this step does not solve the problem, do the rest of the steps in this procedure. 3. See "Performing a preliminary search for a matching problem" on page 41. If you do not find a matching problem, you will need to collect problem information to report to IBM, using the steps in the rest of this list. 4. If the abend did not generate a system dump, and the problem is reproducible, obtain a system dump. See "Obtaining a system dump" on page 47. 5. Format the system dump. See "Formatting a system dump" on page 47. 6. You might want to re-create the problem you are investigating (or set up the conditions to allow the problem to recur) after setting the trap to obtain the dump. 7. See "Searching for a matching problem" on page 82. 8. If you still do not find a matching problem, see Chapter 3, "Reporting RACF problems to IBM," on page 97.
Abends: X'482' X'483' X'485'	`Description:` RACF manager error during SVC (supervisor call instruction) processing. `Action:` 1. See the message issued with the abend. This could be message IRR401I or ICH409I. **Find the abend reason code, if any, in the text for the messages.** 2. See message IRR413I, which occurs with RACF manager abends. Find the RACF manager request ID in the message. 3. See *z/OS Security Server RACF Messages and Codes* for information about the abend code and the accompanying message (if you have not looked up the explanation already). In general, the information in *z/OS Security Server RACF Messages and Codes* should be sufficient to correct the problem. 4. If the information in *z/OS Security Server RACF Messages and Codes* is not sufficient to correct the problem, do the rest of the steps in this procedure. 5. See "Performing a preliminary search for a matching problem" on page 41. If you do not find a matching problem, you need to collect problem information to report to IBM, using the steps in the rest of this list. 6. If the abend did not generate a system dump, and the problem is reproducible, obtain a system dump. See "Obtaining a system dump" on page 47. 7. Format the system dump. See "Formatting a system dump" on page 47. 8. You might want to re-create the problem you are investigating (or set up the conditions to allow the problem to recur) after setting the trap to obtain the dump. 9. See "Searching for a matching problem" on page 82. 10. If you still do not find a matching problem, see Chapter 3, "Reporting RACF problems to IBM," on page 97.

Table 3. Troubleshooting table: Abends (continued)	
Symptom	**Recommended procedures for Abends**
Abends: X'4C6'	`Description:` A service call to a RACF module used for z/OS UNIX services did not complete. The specific error is indicated by the abend reason code. `Action:` 1. See the message issued with the abend. This could be message ICH409I. **Find the abend reason code, if any, in the text for the message.** 2. If you have not looked up the explanations already, see *z/OS Security Server RACF Messages and Codes* for the explanations of the message and the abend code. (Abend codes are also called "completion codes.") **Pay particular attention to the reason code associated with the abend.** In general, the information in *z/OS Security Server RACF Messages and Codes* should be sufficient to correct the problem. 3. See Chapter 3, "Reporting RACF problems to IBM," on page 97.
Abends: X'4C7'	`Description:` RACF storage manager error during processing of a non-SVC, task mode, get space and free space request. `Action:` 1. See the message issued with the abend. This could be message ICH409I. **Find the abend reason code, if any, in the text for the message.** 2. If you have not looked up the explanations already, see *z/OS Security Server RACF Messages and Codes* for the explanations of the message and the abend code. (Abend codes are also called "completion codes.") **Pay particular attention to the reason code associated with the abend.** In general, the information in *z/OS Security Server RACF Messages and Codes* should be sufficient to correct the problem. 3. See Chapter 3, "Reporting RACF problems to IBM," on page 97.

Table 3. Troubleshooting table: Abends (continued)	
Symptom	**Recommended procedures for Abends**
Abends: X'582' X'585'	`Description:` Error during SVC (supervisor call instruction) processing. The specific error is indicated by the abend reason code. `Action:` 1. See message IRR401I or ICH409I, which occur with RACF abends. Find the abend reason code, if any, in the message text. If the abend reason code is zero, no ACEE (accessor environment element) could be found. **Note:** This is normal if a job started or a user logged on while RACF was inactive but has since been reactivated. 2. If the abend reason code is not zero, do the rest of the steps in this procedure. 3. If you have not looked up the explanations already, see *z/OS Security Server RACF Messages and Codes* for the explanations of the message and the abend code. (Abend codes are also called "completion codes.") **Pay particular attention to the reason code associated with the abend.** In general, the information in *z/OS Security Server RACF Messages and Codes* should be sufficient to correct the problem. 4. If the information in *z/OS Security Server RACF Messages and Codes* is not sufficient to correct the problem, do the rest of the steps in this procedure. 5. Check whether any other error messages were issued with the abend messages. These messages might indicate the context in which the abend occurred. In particular, messages preceding the abend message might indicate conditions that caused the abend. **Note:** If message IDs were not included with the messages and you can re-create the error, do: • If a batch job produced the output, check that the job statement had MSGLEVEL=(1,1) specified. • If a TSO/E user had the problem, check that the user profile had these options: PROFILE WTPMSG MSGID.

Table 3. Troubleshooting table: Abends (continued)	
Symptom	**Recommended procedures for Abends**
Abends: X'582' X'585' (continued)	1. Find the name of the **calling** module. This is the module that issued the RACF request (such as a RACF macro, command, or utility). The conditions set up by this module might have caused the abend. See "Identifying the caller of RACF" on page 79. 2. Find the owner of the module: an IBM product, a non-IBM product, or a user at your installation. If the module name begins with ICH or IRR, it is a RACF module. Continue with this procedure. If the module name begins with the letters A through I, S, or X (but not ICH or IRR), it is probably part of an IBM product, but it is not RACF. See the diagnosis or logic document for the program of which the module is a part. To determine the name of the component that owns the module. use the component summary in *z/OS MVS Diagnosis: Reference*. 3. See "Performing a preliminary search for a matching problem" on page 41. If you find a matching problem, apply the appropriate PTFs (program temporary fixes). If you do not find a matching problem, you need to collect problem information to report to IBM, using the steps in the rest of this procedure. **Please be sure to fill out a blank copy of the worksheet shown in Table 20 on page 97.** 4. If the abend did not generate a system dump, and the problem is reproducible, obtain a system dump. See "Obtaining a system dump" on page 47. 5. Format the system dump. See "Formatting a system dump" on page 47. 6. You might want to re-create the problem you are investigating (or set up the conditions to allow the problem to recur) after setting the trap to obtain the dump. 7. See "Searching for a matching problem" on page 82. 8. If you still do not find a matching problem, see Chapter 3, "Reporting RACF problems to IBM," on page 97.

Table 3. Troubleshooting table: Abends (continued)

Symptom	Recommended procedures for Abends
Abends: X'683' X'684' X'685'	`Description:` Insufficient authority to issue the SVC (supervisor call) instruction. The issuer of the SVC must be APF-authorized (authorized program facility). These abends can occur if a RACF command is not in the TSO command table. `Action:` 1. See the message issued with the abend. This could be message IRR401I or ICH409I. **Find the abend reason code, if any, in the text for the messages.** 2. If you have not looked up the explanations already, see *z/OS Security Server RACF Messages and Codes* for the explanations of the message and the abend code. (Abend codes are also called "completion codes.") **Pay particular attention to the reason code associated with the abend.** In general, the information in *z/OS Security Server RACF Messages and Codes* should be sufficient to correct the problem. 3. If the information in *z/OS Security Server RACF Messages and Codes* is not sufficient to correct the problem, do the rest of the steps in this procedure. 4. Check whether any other error messages were issued with the abend messages. These messages might indicate the context in which the abend occurred. In particular, messages preceding the abend message might indicate conditions that caused the abend. **Note:** If message IDs were not included with the messages and you can re-create the error, do: • If a batch job produced the output, check that the job statement had MSGLEVEL=(1,1) specified. • If a TSO/E user had the problem, check that the user profile had these options: PROFILE WTPMSG MSGID. 5. Find the name of the **calling** module. This is the module that issued the RACF request (such as a RACF macro, command, or utility). The conditions set up by this module might have caused the abend. See "Identifying the caller of RACF" on page 79.

Table 3. Troubleshooting table: Abends (continued)	
Symptom	**Recommended procedures for Abends**
Abends: X'683' X'684' X'685' (continued)	1. Find the owner of the module: an IBM product, a non-IBM product, or a user at your installation. If the module name begins with ICH or IRR, it is a RACF module. Continue with this procedure. If the module name begins with the letters A through I, S, or X (but not ICH or IRR), it is probably part of an IBM product, but it is not RACF. See the diagnosis or logic document for the program of which the module is a part. To determine the name of the component that owns the module, use the component summary in *z/OS MVS Diagnosis: Reference*. 2. See "Performing a preliminary search for a matching problem" on page 41. If you find a matching problem, apply the appropriate PTFs (program temporary fixes). If you do not find a matching problem, you need to collect problem information to report to IBM, using the steps in the rest of this procedure. **Please be sure to fill out a blank copy of the worksheet shown in Table 20 on page 97.** 3. If the SVC (supervisor call instruction) was issued because a user issued a RACF command, check to be sure the RACF command is in the list of APF-authorized (authorized program facility) commands or in the ISPF command tables. 4. See "Performing a preliminary search for a matching problem" on page 41. If you do not find a matching problem, you will need to collect problem information to report to IBM, using the steps in the rest of this list. 5. If the abend did not generate a system dump, and the problem is reproducible, obtain a system dump. See "Obtaining a system dump" on page 47. 6. Format the system dump. See "Formatting a system dump" on page 47. 7. You might want to re-create the problem you are investigating (or set up the conditions to allow the problem to recur) after setting the trap to obtain the dump. 8. See "Searching for a matching problem" on page 82. 9. If you still do not find a matching problem, see Chapter 3, "Reporting RACF problems to IBM," on page 97.

Table 3. Troubleshooting table: Abends (continued)	
Symptom	**Recommended procedures for Abends**
Abends: X'9C7'	`Description:` RACF token error during RACROUTE functions. `Action:` 1. See the message issued with the abend. **Find the abend reason code, if any, in the text for the messages.** 2. See *z/OS Security Server RACF Messages and Codes* for information about the abend code and the accompanying message (if you have not looked up the explanation already). In general, the information in *z/OS Security Server RACF Messages and Codes* should be sufficient to correct the problem. 3. If the information in *z/OS Security Server RACF Messages and Codes* is not sufficient to correct the problem, do the rest of the steps in this procedure. 4. See "Performing a preliminary search for a matching problem" on page 41. If you do not find a matching problem, you need to collect problem information to report to IBM, using the steps in the rest of this list. 5. If the abend did not generate a system dump, and the problem is reproducible, obtain a system dump. See "Obtaining a system dump" on page 47. 6. Format the system dump. See "Formatting a system dump" on page 47. 7. You might want to re-create the problem you are investigating (or set up the conditions to allow the problem to recur) after setting the trap to obtain the dump. 8. See "Searching for a matching problem" on page 82. 9. If you still do not find a matching problem, see Chapter 3, "Reporting RACF problems to IBM," on page 97.
Abends: X'AC5'	`Description:` Internal RACF problem. `Action:` 1. See "Performing a preliminary search for a matching problem" on page 41. If you do not find a matching problem, you need to collect problem information to report to IBM, using the steps in the rest of this list. 2. Format the system dump. See "Formatting a system dump" on page 47. 3. See "Searching for a matching problem" on page 82. 4. If you still do not find a matching problem, see Chapter 3, "Reporting RACF problems to IBM," on page 97.

Table 3. Troubleshooting table: Abends (continued)

Symptom	Recommended procedures for Abends
Abends: X'D82' X'D83' X'D84' X'D85'	`Description:` RACF storage manager error during SVC processing. `Action:` 1. See the message issued with the abend. This could be message IRR401I or ICH409I. **Find the abend reason code, if any, in the text for the messages.** 2. If you have not looked up the explanations already, see *z/OS Security Server RACF Messages and Codes* for the explanations of the message and the abend code. (Abend codes are also called "completion codes.") **Pay particular attention to the reason code associated with the abend.** In general, the information in *z/OS Security Server RACF Messages and Codes* should be sufficient to correct the problem. 3. If the information in *z/OS Security Server RACF Messages and Codes* is not sufficient to correct the problem, do the rest of the steps in this procedure. 4. Check whether any other error messages were issued with the abend messages. These messages might indicate the context in which the abend occurred. In particular, messages preceding the abend message might indicate conditions that caused the abend. **Note:** If message IDs were not included with the messages and you can re-create the error, do: • If a batch job produced the output, check that the job statement had MSGLEVEL=(1,1) specified. • If a TSO/E user had the problem, check that the user profile had these options: PROFILE WTPMSG MSGID. 5. Find the name of the **calling** module. This is the module that issued the RACF request (such as a RACF macro, command, or utility). The conditions set up by this module might have caused the abend. See "Identifying the caller of RACF" on page 79.

Table 3. Troubleshooting table: Abends (continued)	
Symptom	**Recommended procedures for Abends**
Abends: X'D82' X'D83' X'D84' X'D85' (continued)	1. Find the owner of the module: an IBM product, a non-IBM product, or a user at your installation. If the module name begins with ICH or IRR, it is a RACF module. Continue with this procedure. If the module name begins with the letters A through I, S, or X (but not ICH or IRR), it is probably part of an IBM product, but it is not RACF. See the diagnosis or logic document for the program of which the module is a part. To determine the name of the component that owns the module, use the component summary in *z/OS MVS Diagnosis: Reference*. 2. See "Performing a preliminary search for a matching problem" on page 41. If you find a matching problem, apply the appropriate PTFs (program temporary fixes). If you do not find a matching problem, you need to collect problem information to report to IBM, using the steps in the rest of this procedure. **Please be sure to fill out a blank copy of the worksheet shown in Table 20 on page 97.** 3. If the abend did not generate a system dump, and the problem is reproducible, obtain a system dump. See "Obtaining a system dump" on page 47. 4. Format the system dump. See "Formatting a system dump" on page 47. 5. You may want to re-create the problem you are investigating (or set up the conditions to allow the problem to recur) after setting the trap to obtain the dump. 6. See "Searching for a matching problem" on page 82. 7. If you still do not find a matching problem, see Chapter 3, "Reporting RACF problems to IBM," on page 97.
Abends: X'E82' X'E83' X'E84' X'E85'	`Description:` RACF is not installed. `Action:` Make sure that RACF is properly installed.

Documentation troubleshooting table

Table 4. Troubleshooting table: Documentation	
Symptom	**Recommended Procedures for Documentation**
Error in a publication that documents RACF	`Action:` 1. If you have a problem with RACF documentation, see "How to send your comments to IBM" on page xvii to report the documentation error. Report the problem to the IBM support center for these situations: • The correction to the documentation is needed to prevent a severe problem. • You are not sure if the error is in the documentation or the product. • The error is in an online panel. 2. If you plan to report the problem to the IBM support center, see: a. "Performing a preliminary search for a matching problem" on page 41 b. "Searching for a matching problem" on page 82 c. Chapter 3, "Reporting RACF problems to IBM," on page 97

Incorrect output troubleshooting table

This topic describes incorrect output from RACF. Generally, incorrect output can be missing records, duplicate records, incorrect values, format errors, or meaningless data.

Incorrect output could also be the improper issuing of an error message, an abend code, a return code from a macro, or a wait-state code. For example, an abend could be issued when no abend is needed, or a message that should be issued is not issued. Another example is message text that includes incorrect information.

Table 5. Troubleshooting table: Incorrect output	
Symptom	**Recommended procedures for incorrect output**
Incorrect output: Improper access to a protected resource **with no message issued**	`Action:` 1. Check whether any error messages were issued for the user request and make sure that message IDs are included with the messages. If a batch job produced the output, check that the job statement had MSGLEVEL=(1,1) specified. If a TSO/E user had the problem, check that the user profile had these options: PROFILE WTPMSG MSGID. 2. Check the global access checking table for an entry describing the resource. This step does not apply for z/OS or OS/390® z/OS UNIX System Services. 3. Check for an installation exit for RACROUTE REQUEST=AUTH. This step does not apply for z/OS or OS/390 z/OS UNIX System Services. 4. Check the kind of RACROUTE request that was made. Compare it with the level of protection of the resource. (Use LISTDSD or RLIST command for the resource; use LISTUSER command for the user.) This step does not apply for z/OS or OS/390 z/OS UNIX System Services. 5. If the cause of the problem is not apparent, attempt to duplicate the problem with another user doing the same kind of request. Try another user with similar attributes and group associations. If the problem can be duplicated, then determine which user attributes or group associations cause the problem. This can help you define the problem. If the problem cannot be duplicated with other users, try to duplicate the problem with other resources protected like the original. 6. See “Performing a preliminary search for a matching problem” on page 41. If you do not find a matching problem, you need to collect problem information to report to IBM, using the procedures in the remainder of this list. 7. Get a printed or online copy of the incorrect output. You need it to analyze the problem or report the problem to IBM. 8. See “Searching for a matching problem” on page 82. 9. See Chapter 3, “Reporting RACF problems to IBM,” on page 97.

Table 5. Troubleshooting table: Incorrect output (continued)

Symptom	Recommended procedures for incorrect output
Incorrect output: Messages	`Action:` 1. Make sure that you have all the error messages that were issued for the user request, and make sure that message IDs are included with the messages. If a batch job produced the output, check that the job statement had MSGLEVEL=(1,1) specified. If a TSO/E user had the problem, check that the user profile had these options: PROFILE WTPMSG MSGID. 2. Decide which of these types of messages applies: a. Macro error messages (MNOTEs) b. System error messages issued by RACF (messages with prefixes of CST, HCP, ICH, IRR, and RPI) 3. Decide what the nature of the message error is. a. Was the message valid but unexpected? See “Messages troubleshooting table” on page 36. b. Was the message itself incorrect; for example, did it contain incorrect or unreadable data, or was an error message issued for a non-error situation? 4. If you had incorrect output from a RACF macro, see “Utilities” on page 3 for a list of common application programming errors for RACF; the results of some of these errors could appear to be incorrect output from RACF. 5. See “Performing a preliminary search for a matching problem” on page 41. If you do not find a matching problem, you need to collect problem information to report to IBM using the procedures in the remainder of this list. 6. Get a printed or online copy of the incorrect output. You need it to analyze the problem or report the problem to IBM. a. For macro error messages (MNOTEs): Get a copy of the assembler listing that contained the macro error message. b. For system error messages issued by RACF: Get a copy of the part of the job log or system log that contained the message. (The IBM support center might also need to check messages that preceded the error message in the log.) 7. See “Searching for a matching problem” on page 82. 8. See Chapter 3, “Reporting RACF problems to IBM,” on page 97.

Table 5. Troubleshooting table: Incorrect output (continued)

Symptom	Recommended procedures for incorrect output
Incorrect output: Incorrect abends	`Action:` 1. Check whether any error messages were issued in response to the user request and make sure that message IDs are included with the messages. If a batch job produced the output, check that the job statement had MSGLEVEL=(1,1) specified. If a TSO/E user had the problem, check that the user profile had these options: PROFILE WTPMSG MSGID. 2. Decide what the nature of the abend error is. a. Was the abend valid but unexpected? Go to "Abend troubleshooting table" on page 9. b. Was the abend not documented? Go to "Documentation troubleshooting table" on page 23. c. Was the abend itself incorrect? For example, was the abend issued for a non-error situation? d. If you had incorrect output from a RACF macro, see "Utilities" on page 3 for a list of common application programming errors for RACF; the results of some of these errors could appear to be incorrect output from RACF. 3. See "Performing a preliminary search for a matching problem" on page 41. If you do not find a matching problem, you need to collect problem information to report to IBM, using the procedures in the remainder of this list. 4. Get a printed or online copy of the incorrect output. You need it to analyze the problem or report the problem to IBM. **To Get the Dump Output and Analysis Results:** Use the procedures described: a. "Obtaining a system dump" on page 47, if you need to request a dump b. "Formatting a system dump" on page 47 c. "Validating input" on page 79. 5. See "Searching for a matching problem" on page 82. 6. See Chapter 3, "Reporting RACF problems to IBM," on page 97.

Table 5. Troubleshooting table: Incorrect output (continued)	
Symptom	**Recommended procedures for incorrect output**
Incorrect output: Improperly formatted output from RACF LIST commands (LISTDSD, LISTGRP, LISTUSER, RLIST, and RACLINK LIST)	`Action:` 1. Check whether any error messages were issued for the user request and make sure that message IDs are included with the messages. If a batch job produced the output, check that the job statement had MSGLEVEL=(1,1) specified. If a TSO/E user had the problem, check that the user profile had these options: PROFILE WTPMSG MSGID. 2. Decide what caused the incorrect output: • The data read by the command (the RACF database). If this is the case, see the table entry for the RACF database. • RACF's processing of the command request 3. If the database should have been updated by a command that was issued on another system in the RACF remote sharing facility (RRSF) network, check the status of the connection with a TARGET LIST on this system and the originating system. Also, check the RRSFDATA profiles on the originating system. 4. See “Performing a preliminary search for a matching problem” on page 41. If you do not find a matching problem, you need to collect problem information to report to IBM, using the procedures in the remainder of this list. 5. Get a printed or online copy of the incorrect output. You need it to analyze the problem or report the problem to IBM. You can capture the output by executing the command in a batch job, or under the session manager, or in a CLIST in which you trap the output from the command. 6. See “Searching for a matching problem” on page 82. 7. See Chapter 3, “Reporting RACF problems to IBM,” on page 97.

Table 5. Troubleshooting table: Incorrect output (continued)

Symptom	Recommended procedures for incorrect output
Incorrect output: Improperly formatted reports (from DSMON or RACF report writer)	`Action:` 1. Check whether any error messages were issued for the user request and make sure that message IDs are included with the messages. If a batch job produced the output, check that the job statement had MSGLEVEL=(1,1) specified. If a TSO/E user had the problem, check that the user profile had these options: PROFILE WTPMSG MSGID. 2. See "Performing a preliminary search for a matching problem" on page 41. If you do not find a matching problem, you need to collect problem information to report to IBM, using the procedures in the remainder of this list. 3. Get a printed or online copy of the improperly formatted report. You need it to analyze the problem or report the problem to IBM. 4. Check whether the problem is caused by RACF in generating the report or by bad data used as input to the report: As a check on the RACF report writer, you can print (or display) the SMF (system management facility) records using the access method services PRINT command with DUMP specified. As a check on DSMON, use the RACF LIST commands to check the profiles that seem to be reported incorrectly by DSMON. 5. See "Searching for a matching problem" on page 82. 6. See Chapter 3, "Reporting RACF problems to IBM," on page 97.

Table 5. Troubleshooting table: Incorrect output (continued)

Symptom	Recommended procedures for incorrect output
Incorrect output: SMF records (types 80, 81, or 83)	`Action:` 1. Check whether any error messages were issued for the user request and make sure that message IDs are included with the messages. If a batch job produced the output, check that the job statement had MSGLEVEL=(1,1) specified. If a TSO/E user had the problem, check that the user profile had these options: PROFILE WTPMSG MSGID. 2. If the RACF SMF data unload utility leads you to suspect that the SMF (system management facilities) records are not valid, check to make sure that the SMF records are at fault, not the utility. 3. See "Performing a preliminary search for a matching problem" on page 41. If you do not find a matching problem, you need to collect problem information to report to IBM, using the procedures in the remainder of this list. 4. Get a printed or online copy of the incorrect output. You need it to analyze the problem or report the problem to IBM. For SMF records (type 80 or 81): • Use the RACF SMF data unload utility to list the contents of the RACF SMF records. See *z/OS Security Server RACF Auditor's Guide* for more information. • As a check on the RACF SMF data unload utility, you can print (or display) the record using the access method services PRINT command with HEX specified. • See "Searching for a matching problem" on page 82. • See Chapter 3, "Reporting RACF problems to IBM," on page 97.

Table 5. Troubleshooting table: Incorrect output (continued)

Symptom	Recommended procedures for incorrect output
Incorrect output: The RACF database (this includes profiles, index blocks, and inventory control block)	`Action:` 1. Check whether any error messages were issued for the user request and make sure that message IDs are included with the messages. If a batch job produced the output, check that the job statement had MSGLEVEL=(1,1) specified. If a TSO/E user had the problem, check that the user profile had these options: PROFILE WTPMSG MSGID. 2. These RACF functions affect the RACF database: • RACF commands that add, alter, or delete profiles (such as ADDUSER, ALTUSER, DELUSER, RDEFINE, RALTER, and RDELETE) • BLKUPD command • RVARY and SETROPTS commands • IRRUT400 utility • RACF macros: RACDEF, RACINIT, ICHEINTY, ICHETEST, ICHEACTN, and RACXTRT (with ENCRYPT or TYPE=REPLACE specified) 3. If the database should have been updated by a command that was issued on another system in the RACF remote sharing facility (RRSF) network, check the status of the connection with a TARGET LIST on this system and the originating system. Also, check the RRSFDATA profiles on the originating system. 4. If you had incorrect output from a RACF macro, see "Utilities" on page 3 for a list of common application programming errors for RACF; the results of some of these errors could appear to be incorrect output from RACF. 5. See "Performing a preliminary search for a matching problem" on page 41. If you do not find a matching problem, you need to collect problem information to report to IBM, using the procedures in the remainder of this list. 6. Get a printed or online copy of the incorrect output. You need it to analyze the problem or report the problem to IBM. Run the IRRUT200 utility and the BLKUPD command to get a description of the RACF database. See *z/OS Security Server RACF System Programmer's Guide* for a description of how to use IRRUT200, and "Block update command for a RACF database (BLKUPD)" on page 120 for a description of how to use BLKUPD. 7. See "Searching for a matching problem" on page 82. 8. See Chapter 3, "Reporting RACF problems to IBM," on page 97.

Table 5. Troubleshooting table: Incorrect output (continued)

Symptom	Recommended procedures for incorrect output
Incorrect output: RACF subsystem	`Action:` 1. Check for any messages that might have been issued by the RACF subsystem. 2. Refer to *z/OS Security Server RACF Messages and Codes*. 3. If the RVARY command, along with other commands, is not working, then the subsystem is also not working correctly. 4. Check to see if the subsystem initialized correctly; if it did not, check for an install problem and get a system console log (SYSLOG). 5. If a *userid*.RRSFLIST data set is missing output, or contains incorrect output: • Issue the SET LIST command and verify that: – Automatic direction or password synchronization is active for the type of request for which output is missing. – The OUTPUT keyword specifies the user ID that should be receiving the missing output. – The correct level of output is specified (ALWAYS, WARN, or FAIL). • Check the RRSFDATA profiles and access lists and make sure that the requests are really being directed. • Check the TSO PREFIX specification for the user ID that is missing the output. If it is set to something other than the user ID, the RRSFLIST data set might have a different name than you expect. • Issue the TARGET LIST command on both nodes involved, and make sure that they are both operative-active, and that requests are not queued in one of the INMSG or OUTMSG data sets. • Check whether anyone stopped the RACF subsystem address space, or used the TARGET command to purge an INMSG or OUTMSG file while a request was active. These actions can cause requests and output to be lost. • Check for TSO TRANSMIT messages to the user ID that is missing the output. If there was an error processing the RRSFLIST data set a message should have been sent. Check the console log too. • Make sure that the RRSFLIST data set is allocated with the proper DCB parameters and is not full. If in doubt, you can delete or rename it and let RACF create another one. 6. If an abend occurred, get a copy of the system dump that was taken. 7. If the abend did not generate a system dump, and the problem is reproducible, obtain a system dump. See “Obtaining a system dump” on page 47. 8. Format the system dump. See “Formatting a system dump” on page 47. **Action**

Table 5. Troubleshooting table: Incorrect output (continued)	
Symptom	**Recommended procedures for incorrect output**
Incorrect output: Return or reason code from a RACF macro or utility	`Action:` 1. Check whether any error messages were issued for the user request and make sure that message IDs are included with the messages. If a batch job produced the output, check that the job statement had MSGLEVEL=(1,1) specified. If a TSO/E user had the problem, check that the user profile had these options: PROFILE WTPMSG MSGID. 2. Decide if the return or reason code is: • Incorrect • Undocumented. See “Documentation troubleshooting table” on page 23. 3. See “Utilities” on page 3 for a list of common application programming errors for RACF; the results of some of these errors could appear to be incorrect output from RACF. 4. See “Performing a preliminary search for a matching problem” on page 41. If you do not find a matching problem, you need to collect problem information to report to IBM, using the procedures in the remainder of this list. 5. Get a printed or online copy of the incorrect output. You need it to analyze the problem or report the problem to IBM. **Note:** If the macro return code or reason code was incorrect, request a system dump from a RACF macro request that produced the incorrect output when RACF returns control. **To Get the Dump Output and Analysis Results:** Use the procedures described in these areas: a. “Obtaining a system dump” on page 47, if you need to request a dump b. “Formatting a system dump” on page 47 c. “Validating input” on page 79. 6. See “Searching for a matching problem” on page 82. 7. See Chapter 3, “Reporting RACF problems to IBM,” on page 97.

Logon problems troubleshooting table

Table 6. Troubleshooting table: Logon problems

Symptom	Recommended procedures for logon problems
Users cannot log on	`Action:` 1. Check whether any error messages were issued for the user request and make sure that message IDs are included with the messages. If a batch job produced the output, check that the job statement had MSGLEVEL=(1,1) specified. If a TSO/E user had the problem, check that the user profile had these options: PROFILE WTPMSG MSGID. 2. If only one or a few users cannot log on, check that they are using the current password. If no users can log on, do the rest of this procedure. 3. There might be a password synchronization problem, which you should be able to recognize because the user's old password is either accepted as correct or as expired. Enter a RACLINK LIST command on each system to check the user's user ID association if that's what is being used to synchronize passwords. 4. If the database should have been updated by a command that was issued on another system in the RACF remote sharing facility (RRSF) network, check the status of the connection with a TARGET LIST on this system and the originating system. Also, check the RRSFDATA profiles on the originating system. 5. Check the system date in effect for the current system IPL. If the wrong date is in effect, users who cannot log on might have the REVOKE attribute in their user profiles. See *z/OS Security Server RACF Security Administrator's Guide* for more information about the REVOKE attribute. 6. If you are sharing your database with other systems, make sure that all systems sharing the database are using the same password authentication algorithm, and that there has not been a period of time when the systems were using different algorithms. The default password authentication algorithm for the RACF component of the SecureWay Security Server is the Data Encryption Standard (DES) algorithm. The default password authentication algorithm for RACF on MVS beginning with RACF 2.1 is the (DES) algorithm, but for releases previous to RACF 2.1 the default is the masking algorithm. Make sure that you have taken the correct steps to activate the algorithm that you intend to use on each system sharing the database. See *z/OS Security Server RACF System Programmer's Guide* for more information.

Table 6. Troubleshooting table: Logon problems (continued)	
Symptom	**Recommended procedures for logon problems**
Users cannot log on (continued)	1. If you are using a PassTicket, make sure: • The PTKTDATA class has been activated and a class profile exists. • You do not try to use the same PassTicket more than once. • The GMT clock on the evaluating computer is within the valid time range. For more information, see Using PassTickets in *z/OS Security Server RACF Security Administrator's Guide*. 2. See “Performing a preliminary search for a matching problem” on page 41. If you do not find a matching problem, you need to collect problem information to report to IBM, using the procedures in the remainder of this list. 3. Get a printed or online copy of the SMF records related to the logon problem. You need it to analyze the problem or report the problem to IBM. 4. See “Searching for a matching problem” on page 82. 5. See Chapter 3, “Reporting RACF problems to IBM,” on page 97.

Loops troubleshooting table

Table 7. Troubleshooting table: Loops

Symptom	Recommended procedures for loops
Any loop during RACF processing	`Action:` 1. Check whether any error messages were issued for the user request that had the loop and make sure that message IDs are included with the messages. If a batch job produced the output, check that the job statement had MSGLEVEL=(1,1) specified. If a TSO/E user had the problem, check that the user profile had these options: PROFILE WTPMSG MSGID. 2. Check whether the prefix of one of the modules involved in the loop is ICH or IRR. **Note:** The instructions involved in the loop are generally pointed to by the PSW that was active when the system dump was generated. Most modules owned by IBM have an identifier with the module name at either the beginning or end of the object code for the module. See the MVS diagnosis publications if you need further details of general loop analysis techniques. 3. If the prefix is ICH or IRR, the problem might be caused by RACF, and you should use these diagnostic procedures: a. Check that the problem was not due to repeated requests for RACF services. Repeated requests might be caused by a loop of logic in the application program issuing the request. If the loop was in a program making repeated requests, report the problem to the owner of the program. If the loop was within RACF, use the diagnostic procedures in the remainder of this list. b. See “Performing a preliminary search for a matching problem” on page 41. c. If you do not find a matching problem, you need to request a dump (if you don't have one already) and collect problem information to report to IBM. **To Get the Dump Output and Analysis Results**: Use the procedures described in these areas: i) “Obtaining a system dump” on page 47, if you need to request a system dump. ii) “Formatting a system dump” on page 47. iii) “Validating input” on page 79. d. See “Searching for a matching problem” on page 82. e. See Chapter 3, “Reporting RACF problems to IBM,” on page 97. 4. If the prefix of one of the looping modules is **neither ICH nor IRR**, the problem was probably not caused by RACF. Note the prefixes of the looping modules. If the module is owned by IBM (modules owned by IBM usually start with the letters A through I, S, or X), see the diagnosis or logic document for the program that the module is a part of. To determine the name of the component that owns the module, use the component summary in *z/OS MVS Diagnosis: Reference*.

Messages troubleshooting table

Table 8. Troubleshooting table: Messages

Symptom	Recommended procedures for messages
Messages with a prefix of IKJ	`Description:` These are TSO messages issued for RACF commands. IKJ messages are issued by the TSO Terminal Monitor Program (TMP). `Action:` 1. See *z/OS TSO/E Messages* for a description of the message and the condition causing it. Attempt to correct the problem from the information in the messages document. 2. If you cannot correct the problem from the information in the messages document, do the problem determination procedure related to the message.
Messages with a prefix of ICH or IRR	`Description:` These are RACF messages. `Action:` 1. See *z/OS Security Server RACF Messages and Codes* for a description of the message and the condition causing it. Attempt to correct the problem from the information in *z/OS Security Server RACF Messages and Codes*. 2. If you cannot correct the problem from the information in *z/OS Security Server RACF Messages and Codes*, do the problem determination procedure related to the message.
Messages with a prefix of BPX	`Description:` These are z/OS and OS/390 z/OS UNIX System Services messages. `Action:` If there are no RACF messages with this message, the problem is probably not caused by RACF. See your system diagnosis guide for information about correcting this problem.
Unnumbered messages	`Description:` Unnumbered messages are also issued by RACF. `Action:` 1. Make sure that any message IDs are included with the messages. If a batch job produced the output, check that the job statement had MSGLEVEL=(1,1) specified. If a TSO/E user had the problem, check that the user profile had these options: PROFILE WTPMSG MSGID. 2. See *z/OS Security Server RACF Messages and Codes* for a description of the message and the condition causing it. Attempt to correct the problem from the information in the messages document. 3. If the message is not in *z/OS Security Server RACF Messages and Codes*, and you are certain it is a RACF message, see “Documentation troubleshooting table” on page 23.

Performance problems troubleshooting table

Table 9. Troubleshooting table: Performance problems

Symptom	Recommended procedures for performance problems
Performance, such as: • There is excessive I/O to the RACF database. • Users report slow response time, particularly when logging on or off, or when opening data sets.	`Action:` Most performance problems related to RACF are corrected by changing the way RACF is set up at an installation. Before suspecting that RACF itself is causing performance problems, review this checklist to assure that the problem is not actually caused by the way your installation has set up RACF. **Pay close attention to any recent changes you have made in the way you have set up RACF**. For more information about these performance considerations, see *z/OS Security Server RACF System Programmer's Guide*. • The rate at which users log on to TSO/E, sign on the IMS/VS or CICS®/VS, or submit batch jobs on MVS affects system performance. The higher the rate, the lower the performance. You might see peaks and valleys in performance through the work day. • Using the RACLIST or GENLIST operands on the SETROPTS command can improve system performance. • Use of the global access checking table can improve system performance. • Using RACROUTE REQUEST=FASTAUTH instead of RACROUTE REQUEST=AUTH can improve system performance. • Using ADSP and PROTECT=YES can degrade system performance. • The processing performed by installation-written exit routines can affect system performance. • The placement of the RACF database on DASD (including how many DASDs the database is placed on) can affect system performance. • The use of logging by users with SPECIAL or group-SPECIAL attributes or users with AUDITOR or group-AUDITOR attributes can reduce system performance. • The gathering of resource statistics can reduce system performance. • The use of certain RACF commands (such as SEARCH, LISTDSD with the ID or PREFIX operands, LISTGRP *, and others) can reduce system performance when there is a single RACF database. • The use of RACF utilities can reduce system performance when there is a single data set in the RACF database. • The use of erase-on-scratch can reduce system performance.

Table 9. Troubleshooting table: Performance problems (continued)

Symptom	Recommended procedures for performance problems
Performance (continued)	• During failsoft processing, the operator is prompted frequently, thereby degrading system performance while you repair RACF. To improve system performance during failsoft processing, write a RACROUTE REQUEST=AUTH preprocessing exit. • List of groups processing can reduce system performance. • Group authority processing when made to percolate for a user to subgroups owned by the group can reduce system performance. • Defining many generic profiles within a data set high-level qualifier or a general resource class can hurt performance. • The use of the IRRACEE VLF class can improve system performance. The use of the IRRUMAP, IRRGMAP, and IRRSMAP VLF classes can improve performance of z/OS and OS/390 z/OS UNIX System Services. • See "Performing a preliminary search for a matching problem" on page 41. If you do not find a matching problem, you will need to collect problem information to report to IBM, using the procedures in the remainder of this list. • See "Searching for a matching problem" on page 82. • See Chapter 3, "Reporting RACF problems to IBM," on page 97.

RACF return codes troubleshooting table

Table 10. Troubleshooting table: Return codes

Symptom	Recommended procedures for return codes
Return codes from RACF macros (nonzero value in register 15)	`Action:` 1. For some macros, depending on the return code, there is a reason code in register 0. Check register 0 for this reason code. (To obtain the reason code from a RACF macro, see "Obtaining RACF return codes and reason codes from a RACF macro invocation" on page 48.) 2. See "Utilities" on page 3 for a summary of some commonly encountered macro reason codes and advice on how to resolve problems that are due to an error in the program that issued a RACF macro. **Note:** Error return codes and reason codes from a RACF macro are generally due to a problem in the program that issued the macro. 3. If you haven't done so already, look up the explanation of the RACF macro's return and reason codes. See *z/OS Security Server RACF Macros and Interfaces*, except for the RACROUTE macro, which is described in *z/OS Security Server RACROUTE Macro Reference*. **Note:** For the RACROUTE macro, you should also check the first and second words of the SAF parameter list. These are the return and reason codes from the RACF request (usually a RACF macro invocation). 4. Check whether any error messages were issued for the user request, and make sure that message IDs are included with the messages. If a batch job produced the output, check that the job statement had MSGLEVEL=(1,1) specified. If a TSO/E user had the problem, check that the user profile had these options: PROFILE WTPMSG MSGID. 5. The module that issued the macro is either owned by IBM or is a non-IBM product. If the module is owned by IBM (modules owned by IBM usually start with the letters A through I, S, or X), see the diagnosis or logic document for the program of which the module is a part. To determine the name of the component that owns the module. use the component summary in *z/OS MVS Diagnosis: Reference*. 6. If the problem still appears to be caused by RACF, use the diagnostic procedures described in these areas: a. "Performing a preliminary search for a matching problem" on page 41. If you do not find a matching problem, continue diagnosis, using the procedures in the remainder of this list. b. If you need to request a system dump, "Obtaining a system dump" on page 47. c. "Formatting a system dump" on page 47. d. "Validating input" on page 79. e. "Searching for a matching problem" on page 82. f. Chapter 3, "Reporting RACF problems to IBM," on page 97.
RACF manager return codes	`Action:` 1. See *z/OS Security Server RACF Messages and Codes* for a description of the message and the condition causing it. Attempt to correct the problem from the information in the messages document. 2. If you cannot correct the problem from the information in the messages document, perform the problem determination procedure that is related to the message.

Table 10. Troubleshooting table: Return codes (continued)	
Symptom	**Recommended procedures for return codes**
Return codes from RACF callable services	`Action:` 1. Check the parameter list for the SAF return code, the RACF return code, and the RACF reason code. 2. If you have not done so already, look up the explanations of the return codes and reason codes. See *z/OS Security Server RACF Callable Services* and *z/OS Security Server RACF Messages and Codes*.

Waits troubleshooting table

Table 11. Troubleshooting table: Waits	
Symptom	**Recommended procedures for waits**
Wait: • Enabled wait ("hang") in the task that issued a RACF request • Dummy (no work) wait • Wait caused by RACF	`Action:` 1. First, see your system diagnosis guide to check which module **caused** the wait. Try to find out what address space the module was running in. The key question to answer is, "Why was the job or user waiting?" (If a task is waiting for a RACF macro request, RACF might not have caused the wait; another routine could cause the wait if it never released a resource needed by RACF.) Perform the remaining diagnostic steps only if you have determined that RACF is actually the cause of the problem (that is, RACF never released a resource or caused the wait in some other way). 2. Check whether any error messages were issued for the user request that immediately preceded the wait. If a batch job produced the output, check that the job statement had MSGLEVEL=(1,1) specified. If a TSO/E user had the problem, check that the user profile had these options specified: PROFILE WTPMSG MSGID. 3. If the prefix of a module that caused the wait is ICH or IRR, the problem is probably caused by RACF and these diagnostic procedures apply: a. “Performing a preliminary search for a matching problem” on page 41. If you do not find a matching problem, continue diagnosis, using the procedures in the remainder of this list. b. If you do not have a dump, “Obtaining a system dump” on page 47. c. “Formatting a system dump” on page 47. d. “Validating input” on page 79. e. “Searching for a matching problem” on page 82. f. Chapter 3, “Reporting RACF problems to IBM,” on page 97. 4. If the prefix of the waiting module's name is **neither ICH nor IRR,** the problem was probably not caused by RACF. If the module is owned by IBM (modules owned by IBM usually start with the letters A through I, S, or X), see the diagnosis or logic document for the program of which the module is a part. To determine the name of the component that owns the module, use the component summary in *z/OS MVS Diagnosis: Reference*.

Performing a preliminary search for a matching problem

Introduction to searching for a matching problem

This topic discusses how to create a list of symptoms for RACF. You (or IBM) can use these symptoms in the search argument for a preliminary search of a problem database.

A search argument consists of a set of problem symptoms, called search symptoms, to use in a search to identify a matching problem.

If you have a search tool such as Information Management, consider searching these for a matching problem (which could have a fix that is already developed). You should search in the order shown:

1. PSP bucket (Preventive Service Plan)
2. Your problem database (if you have one)
3. An IBM problem database

If you find that the problem has already been reported, you can avoid or reduce time-consuming problem analysis work. In addition, if you have the Info/Access tool, or SoftwareXcel Extended in the ServiceLink application on IBMLink, you can order the fix for the problem.

If you don't have a search tool, you need to collect additional search symptoms before you report the problem to IBM. (The troubleshooting tables list the additional procedures to use.)

If you are doing a structured search, you should provide symptoms in the order that they are listed in DAEDATA and SYMPTOMS output to have the fastest search.

The procedure gives examples of the free-format search symptoms that you can use if you have a search tool such as Info/System (with the Info/MVS database) or Info/Management. It also has examples of the search symptoms that you need if you do a structured search using a tool such as Info/Management or Info/Access. You might need the structured symptoms if you call the IBM support center to report a problem or if you review the APAR and PTF descriptions in IBM's Early Warning System (EWS). EWS is on microfiche and on the Info/MVS database.

The formats used in the examples are the same formats that are used at the IBM support center to enter problem descriptions for APARs and to do searches.

Related information

If you need information about doing problem searches with the Info/System, Info/MVS, Info/Management, or Info/Access program, see *Introducing the Information/Family*.

For examples of Info/Management panels and structured search symptoms to use in a search for a matching problem, see *Using Information/Family Problem Management*.

Steps for searching for a matching problem

Before you begin: You need to verify that this is a problem caused by RACF and to collect some initial problem symptoms. See “Troubleshooting tables for problems in RACF” on page 7 for steps to perform this prerequisite.

Perform these steps to:

1. Make an initial list of search symptoms.
2. Use the list to search for a matching problem.
3. If you did not find a match, try other symptoms.
4. If you found too many matches, add search symptoms.

Details of the procedure are:

1. Use Table 12 on page 42 to select which search arguments apply to the type of problem you have.

---

2. Using Table 13 on page 44 for examples of search arguments, do your first search for a match.

---

3. If your search does not produce a match, try widening your search.
   a. Use the word "RACF" instead of the component ID.
   b. Use fewer search arguments.

---

4. If a search produces too many matches, use some of the other symptoms listed in this procedure to narrow the search.

---

5. If your preliminary searches are unsuccessful, do the structured search described in "Searching for a matching problem" on page 82.
   a. If a structured search is still unsuccessful, gather additional information before reporting the problem to IBM by using the diagnostic procedures recommended for your type of problem.
   b. Return to where you were in the troubleshooting tables.

---

6. If your search has too many matches, consider using these procedures to add search symptoms:
   a. Return to the troubleshooting tables to obtain one or more additional symptoms to narrow your search.
   b. Obtain additional search symptoms by analyzing a dump or doing other in-depth analysis. (For lists of the procedures for doing the analysis, return to where you were in the troubleshooting tables.)

      Note: "Searching for a matching problem" on page 82 summarizes the additional symptoms that you can obtain through in-depth analysis.

You know you are done when you have found a problem match for your symptom or have contacted IBM for additional help.

Table 12. Preliminary search symptoms for RACF

Search Argument	**Free-Format Example**	**Structured Example**
Component identification (ID) The component ID matches problems that had RACF as either the cause of the problem or the apparent cause (for a problem that was reassigned to another component).	RACF 5752XXH00 5752SC1BN	PIDS/5752XXH00 PIDS/5752SC1BN

Table 12. Preliminary search symptoms for RACF (continued)

Search Argument	Free-Format Example	Structured Example
Release level of RACF: Use the FMID (function modification identifier). For a list of RACF FMIDs, see Table 14 on page 45.	R902 or RI02 (R for release; I, the ninth letter of the alphabet, can stand for 9; and 02 for 02. R922 (for RACF 1.9.2) R210 (for RACF 2.1) R220 (for RACF 2.2) R230 (for FMID HRF2230) R240 (for FMID HRF2240) R260 (for FMID HRF2260) 2608 (for FMID HRF2608) 7703 (for FMID HRF7703)	LVLS/902 or LVLS/RI02 LVLS/922 LVLS/210 LVLS/220
RACF request that failed	RACHECK PROFILE	PCSS/RACHECK PCSS/PROFILE
Error reason code from the RACF request (if available)	rc00000004	PRCS/00000004
Associated error message ID The job log or system log might contain error messages associated with the problem that you encountered. Some problems produce multiple error messages, such as problems initiated by a previous abend or I/O error. General error messages, such as the IEA995I symptom dump message for all types of abends, are not needed as search symptoms.	msgICH408I	MS/ICH408I
Associated reason code from message	rc04	PRCS/04
RACF module	ICHRIN00	RIDS/ICHRIN00
SMF record with error	SMF type 80	RECS/SMF80
Part of a document title	Security Administrator's Guide	PUBS/security PUBS/administrator's PUBS/guide
Symptoms for later searches		
Component ID of a service used by RACF, such as Master Scheduler If you suspect a problem in a system service used by RACF, consider a search argument of the service's component ID, plus the failing RACF request and the problem type (such as an abend and reason code).	5752SC1B8	PIDS/5752SC1B8

Table 12. Preliminary search symptoms for RACF (continued)		
Search Argument	**Free-Format Example**	**Structured Example**
A PTF number Consider use of this symptom if a recent PTF (program temporary fix) updated a RACF module involved in the problem. This does not mean that an error exists in the PTF, only that the problem might occur after this PTF is installed.	UY00934 UW02010	PTFS/UY00934 PTFS/UW02010
Area within a document, such as a part, section, chapter, appendix, or page	Chapter3 appendixA	PUBS/CHAP3 PUBS/APPXA
Document order number Document order numbers end with a version number, such as the 01 in GC28-1154-01. All version numbers are now two digits, although previously they could be one. You could leave off the second hyphen and the version numbers that follow it. Then, if you receive too many matches, add the version numbers to the search symptom.	GC28-1154-01	PUBS/GC28115401

Examples of search arguments for RACF

Table 13 on page 44 has examples of free-format search arguments for problems in RACF.

Table 13. Examples of search arguments for RACF		
Problem type	**Example of search argument**	**Descriptions of search symptoms**
Abend	5752XXH00 abend685 rc04	Component ID Abend code Abend reason code (if one exists)
Documentation error	5752XXH00 doc RACHECK	Component ID The word *doc* The name of the RACF request with the documentation error
Incorrect output	5752XXH00 RACHECK output	Component ID The type of output request that was involved The word *output*

Table 13. Examples of search arguments for RACF (continued)

Problem type	Example of search argument	Descriptions of search symptoms
Loop	5752XXH00 ICHRIN00 enabled loop	Component ID The name of any associated module (or RACF request) The word *loop*. Add the word *enabled* or *disabled* if you know the type of loop.
Message	5752XXH00 msgICH408I invalid password	Component ID The message identifier Any inserted code (or text) in the message
Performance degradation	5752XXH00 RACHECK performance	Component ID The name of any associated RACF request The word *performance*
Return code from macro	5752XXH00 RACHECK rc08rc04	Component ID The return code and any reason code (Specify both return code and reason code if a reason code occurred.)
Wait - *hang* or coded wait state (using ICHRIN00)	5752XXH00 ICHRIN00 enabled wait	Component ID The name of any associated RACF module (or RACF request) The word *wait* and any associated wait-state code (for a wait state indicated at the system operator's console). If you know the type of wait, add the word *enabled*, *disabled*, or *dummy* (for a no-work wait).
Wait - *hang* or coded wait state (using RACHECK)	5752XXH00 RACHECK wait03c	Component ID The name of any associated RACF module (or RACF request) The word *wait* and any associated wait-state code (for a wait state indicated at the system operator's console). If you know the type of wait, add the word *enabled*, *disabled*, or *dummy* (for a no-work wait).

Table 14. RACF FMIDs

RACF Level	RACF FMID
RACF 1.9	HRF1902
RACF 1.9.2	JRF1922
RACF 2.1	HRF2210
RACF 2.2	HRF2220
OS/390 Release 1	HRF2220

Table 14. RACF FMIDs (continued)

RACF Level	RACF FMID
OS/390 Release 2	HRF2220
OS/390 Release 3	HRF2230
OS/390 Version 2 Release 4	HRF2240
OS/390 Version 2 Release 5	HRF2240
OS/390 Version 2 Release 6	HRF2260
OS/390 Version 2 Release 7	HRF2260
OS/390 Version 2 Release 8	HRF2608
OS/390 Version 2 Release 9	HRF2608
OS/390 Version 3 Release 10	HRF7703
OS/390 Version 3 Release 12	HRF7705
z/OS Version 1 Release 3	HRF7706
z/OS Version 1 Release 4	HRF7707
z/OS Version 1 Release 5	HRF7708
z/OS Version 1 Release 6	HRF7709
z/OS Version 1 Release 7	HRF7720
z/OS Version 1 Release 8	HRF7730
z/OS Version 1 Release 9	HRF7740
z/OS Version 1 Release 10	HRF7750
z/OS Version 1 Release 11	HRF7760
z/OS Version 1 Release 12	HRF7770
z/OS Version 1 Release 13	HRF7780
z/OS Version 2 Release 1	HRF7790
z/OS Version 2 Release 2	HRF77A0
z/OS Version 2 Release 3	HRF77B0
z/OS Version 2 Release 4	HRF77C0
z/OS Version 2 Release 5	HRF77D0

Chapter 2. Collecting and analyzing problem data

This topic explains how to collect and analyze problem data. These tasks include obtaining dumps and traces, validating input, and searching for matching problems.

Collecting problem data

If you do not know what data to collect for the problem that you are diagnosing, see "Troubleshooting the problem" on page 1. Also, Chapter 3, "Reporting RACF problems to IBM," on page 97 summarizes the information that you should consider collecting before reporting each type of problem to IBM.

Obtaining a description of a RACF installation

For advice on gathering information about a RACF installation, see the information about conducting an audit in *z/OS Security Server RACF Auditor's Guide*.

Obtaining a system dump

Prerequisites

You have a problem with an abend for which a system dump was not taken.

How to obtain a system dump

If the problem occurred while a batch job was using the DSMON command (or the RACF report writer), run the job again with a SYSABEND DD statement specified. This generates a system dump.

If an abend does not produce a system dump, and the problem is reproducible, set a SLIP trap to obtain a system dump the next time the problem recurs.

The trap set by this procedure causes an SVC dump to be generated. Generally, an SVC dump contains all the information needed for debugging RACF-related problems. You may want to re-create the problem you are investigating (or set up the conditions to allow the problem to recur) after setting the trap.

1. On the operator's console, enter this command:

   ```
   SLIP SET,COMP=xxx,ACTION=SYNCSVCD,JOBNAME=jobname,END
   ```

   where *xxx* is the system completion code and *jobname* is the job name (if a batch job is involved) or the user ID (if an interactive user is involved).

   Note: For more information about setting the SLIP trap, see *z/OS MVS System Commands*.
2. When the abend occurs, the system console receives message IEA989I:

   ```
   SLIP TRAP ID=nnn MATCHED
   ```

   where *nnn* is the system-assigned SLIP ID. The operator should notify you if a system dump was requested. You can format and read the system dump.

   Note: A system dump is not generated if the SLIP is a NODUMP SLIP.

Formatting a system dump

Prerequisites

- IPCS
- A system dump (see "Obtaining a system dump" on page 47)

How to format a system dump

To start debugging a system dump, the IBM support center requires this information to be formatted in the dump.

- Use IPCS to format the system dump by using:
  - SUMMARY
  - SUMDUMP
  - FORMAT
  - CURRENT
  - TRACE
  - LOGDATA
  - LPAMAP
  - CPUDATA
  - SYMPTOMS

  This can help you start researching a problem. The IBM support center may request further information, for which other verbexits would be used for further research on specific problems.

Obtaining RACF return codes and reason codes from a RACF macro invocation

If the program that issued the RACF macro received an error return code from the macro, but did not produce a message containing the return and reason codes, you can use one of these techniques to obtain the codes:

- Use the RACF SET TRACE command. See “Obtaining trace records using the SET TRACE command” on page 55.
- Place a TSO TEST trap at the instruction after the macro expansion instructions in the program, and check the return and reason codes when the trap matches.

  Note: You could use the SLIP command to set a trap, but it affects system performance. If you use SLIP, consider qualifying it with job name or PVTMOD options, and request an SVC dump when the trap is hit.
- Use the SPZAP ("superzap") service aid to modify the program that issued the RACF macro:
  1. Locate the RACF macro instruction in the assembly listing or the object code for the program.
  2. Use the SPZAP service aid to change the program's instruction that follows the last instruction of the macro expansion. Change the instruction operation (op) code to an invalid op code, such as X'00'.
  3. Re-create the problem, using JCL with a SYSMDUMP, SYSUDUMP, or SYSABEND DD statement. The invalid op code causes an X'0C1' abend.

     The SYSMDUMP DD statement is recommended if you plan to report the problem to IBM.

Obtaining RACF return codes and reason codes from a RACF callable service invocation

If the program that invoked the callable service received an error return code from the macro, but did not produce a message containing the return code and reason code, you can use one of these techniques to obtain the codes:

- Use the RACF SET TRACE command. See “Obtaining trace records using the SET TRACE command” on page 55.
- Place a TSO TEST trap at the instruction after the callable service invocation in the program, and check the return code and reason code when the trap matches.

Note: You could use the SLIP command to set a trap, but it affects system performance. If you use SLIP, consider qualifying it with job name or PVTMOD options, and request an SVC dump when the trap is hit.

- Use the SPZAP ("superzap") service aid to modify the program that invoked the callable service:
  1. Locate the callable service invocation in the assembly listing or the object code for the program.
  2. Use the SPZAP service aid to change the program's instruction that follows the callable service invocation. Change the instruction operation (op) code to an invalid op code, such as X'00'.
  3. Re-create the problem, using JCL with a SYSMDUMP, SYSUDUMP, or SYSABEND DD statement. The invalid op code causes an X'0C1' abend.

     The SYSMDUMP DD statement is recommended if you plan to report the problem to IBM.

At the time of the trap or the 0C1 abend, Register 1 contains the address of the parameter list for the callable service. This parameter list contains the SAF return code and the RACF return code and reason code. See *z/OS Security Server RACF Callable Services* for more information.

Obtaining traces

When access to a resource is allowed (or denied) incorrectly, you can obtain more data about the problem with GTF trace. GTF trace must be activated to obtain trace output just before reproducing the problem.

To obtain a trace when access is incorrectly denied, do:

1. To start GTF, enter this command:

   ```
   START GTF,TRACE=SLIP
   ```

2. Set this SLIP trap:

   ```
   SLIP SET,IF,LPAEP=(ICHRFR00,0,0),ACTION=TRACE,JOBNAME=xxx,
        TRDATA=(STD,REGS,1R??,+100),END
   ```

   where *xxx* is the job name of a batch job or the user ID of an interactive user.

   This SLIP trap produces a GTF trace entry each time a RACROUTE macro is invoked (ICHRFR00 is the module called by RACROUTE). The trace entry contains the parameter list passed with the RACROUTE macro request.

3. Use the START GTF command to trace the SVCs related to the RACHECK macro (equivalent to SVC 130) and RACDEF macro (equivalent to SVC 133). There is more than one way to do this. This is a suggestion only:

   a. On the master console, enter the START GTF command with TRACE=SVCP specified.

   b. When GTF prompts for trace-event keywords, specify SVC=(130,133).

4. Run the job that has the access problem (or ask the user to attempt to gain access to the resource again) while GTF is on.
5. Stop GTF after the job has ended or after the user has attempted access.
6. Examine the GTF trace output. See *z/OS MVS IPCS Commands* for more information.

Obtaining traces for program control and Program Access to Data Set (PADS) errors

Program control allows an installation to treat load modules (programs) as protected resources. This gives installations the ability to control who can execute which programs.

Load modules are protected by creating a profile for the program in the PROGRAM general resource class. A program protected by such a profile is called a controlled program.

An installation can use a controlled program as a condition for access to a specified data set. That is, some users can access specified data sets at a specified access level only while executing a certain

controlled program. This is known as program access to data sets (PADS). PADS is set up by creating a conditional access list for the data set profile protecting the data sets.

In some cases, users trying to implement program control and PADS might receive message IEC1501I, ABEND 913-38, or message ICH408I, INSUFFICIENT ACCESS AUTHORITY, when they feel they have the necessary authority to open the data set through the conditional access list of the data set. In these scenarios, these error messages most commonly occur when a controlled program loaded an uncontrolled program.

Other error messages are generated when dealing with program control and PADS in the user's environment. These RACF processing messages can be used to diagnose errors when defining programs to RACF and the file system. Check the job log and security console for these diagnostic messages, and use the responses in *z/OS Security Server RACF Messages and Codes* to attempt to correct the problem before setting the slip traps.

In a RACF environment, a program and user combination can open a RACF-protected data set through conditional access list authority if all of the other programs in the environment are RACF-protected. RACF turns on a bit in the TCB, known as the "dirty bit" or TCBNCTL, to indicate that a program not protected by RACF is loaded into the environment. If this bit is on, an uncontrolled program causes a failure. RACF provides an environment service, IRRENS00, to assist in handling program control and PADS. For more information about this service, see *z/OS Security Server RACF Macros and Interfaces*.

Thus, in a RACF environment, it is **not** sufficient to only protect the program doing the open of the data set. To correctly implement PADS, you must protect all of the programs loaded in the environment before the attempt to open the data set. This is also true for any utilities that are to be executed. If a utility loads another program during execution, this additional program must also be protected to prevent an error condition. To help users determine what programs need to be protected, a TRACE is provided in “Trace examples” on page 50 so that when set correctly, it prints out the program name, data set name, and volume required to define profiles in the PROGRAM class for these other programs.

Attention: Read all of the information that follows before beginning any of the activities described.

Trace examples

This information provides two examples of traces. The first example writes a GTF trace record for each program that is loaded into the environment. The second example can determine additional programs that need to be protected.

Trace example 1

This slip should be used to write a GTF trace record for each program loaded into the environment:

```
SLIP SET,IF,ACTION=TRACE,LPAMOD=(ICHRFR00,xxxxx),J=jobname,
TRDATA=(STD,REGS,zzzzzz),ML=100,END
```

xxxxx
The offset into ICHRFR00 where ICHRCP00 starts plus the *yyy* value defined in Table 15 on page 51 for your RMID level of the ICHRCP00 CSECT. Run an AMBLIST of ICHRFR00 to determine the offset of the beginning of ICHRCP00 into ICHRFR00.

zzzzz
See Table 15 on page 51 for the correct TRDATA information based on your RMID level of the ICHRCP00 CSECT. The *yyy* and *zzzzz* values specified for UW21213 should be used for all later RMID levels beginning with RACF 2.2.0, and for the OS/390 Release 3 level of RACF (HRF2230). If you have a later RMID level for RACF 2.1.0, you should use the *yyy* and *zzzzz* values specified for UW16891.

jobname
If the program is executed in batch, this is the name of the job that executes the program. If a TSO user executes the job, this is the TSO user ID. In the case of a TSO user ID, you must set the slip before the user logging on to the system.

Table 15. RMID levels, yyy and zzzzz values for example 1		
RMID LEVEL	**yyy**	**zzzzz**
UW03221	406	9R?+A0?,+7,9R?+A4?,+2B,9R?+A8?,+5
UW08914	406	9R?+A0?,+7,9R?+A4?,+2B,9R?+A8?,+5
UW16891	BC	9R?+A0?,+7,9R?+A4?,+2B,9R?+A8?,+5
HRF2220	5E	9R?+00?,+7,9R?+04?,+2B,9R?+08?,+5
UW21213	60	9R?+00?,+7,9R?+04?,+2B,9R?+08?,+5
HRF2240 to HRF2608	80	9R?+00?,+7,9R?+04?,+2B,9R?+08?,+5
HRF7703	84	9R?+00?,+7,9R?+04?,+2B,9R?+08?,+5
HRF7707	9C	9R?+00?,+7,9R?+04?,+2B,9R?+08?,+5
UA81995 / UA81996 / UA81997	A4	9R?+00?,+7,9R?+04?,+2B,9R?+08?,+5
UA83891 / UA83892 / UA83893	9C	9R?+00?,+7,9R?+04?,+2B,9R?+08?,+5

Because this slip produces GTF records, you need to start GTF ensuring that you use PARM TRACE=SLIP, then use IPCS to format the data with the GTFTRACE IPCS command.

If the preceding steps have been implemented correctly, 'SLIP S+U' entries are generated in the trace records. Under the 'SLIP S+U' heading, locate the 'General Purpose Register Values' line and obtain the value of R15 for each of the 'SLIP S+U' records in the trace output.

R15 can have 5 possible values:

0
User or group is authorized to resource.

4
Resource is not defined to RACF.

8
User or group is not authorized to resource.

C
Resource is not defined to RACF and library is controlled.

10
User or group is authorized to resource and program has NOPADCHK attribute.

If the value in R15 is 4, 8, or C, a profile in the PROGRAM class must be defined to protect the program identified by this trace entry. In addition to defining the profile in the PROGRAM class, the PERMIT command must be issued to put users or groups in the access list for program's profile. To rebuild the in-storage profile list, issue the SETROPTS WHEN(PROGRAM) REFRESH command after making changes to the PROGRAM class. This allows the changes to take effect immediately. For more information about defining profiles in the PROGRAM class and creating entries in a conditional access list, see *z/OS Security Server RACF Security Administrator's Guide*.

Note: If the RACF database is being shared with other systems, the SETROPTS REFRESH takes effect only on the system on which it was issued. In this case, the SETROPTS WHEN(PROGRAM) REFRESH must be issued on all the other sharing systems. This allows the PROGRAM class changes to take effect immediately on the other systems as well. An exception occurs when RACF is enabled for sysplex communications.

The information that you need to define the PROGRAM class profile correctly is found in the 'SLIP USR' of the trace record following the 'SLIP S+U' information. If the *zzzzz* value was set correctly, it is:

```
0008  PROGRAM
002C  DATASET.NAME
0006  VOLUME
```

where:

PROGRAM
Is the actual name of the program being loaded as it is known to RACF.

DATASET.NAME
Is the name of the data set from which the program name was loaded. There are instances where the library name does not show up in the trace.

VOLUME
Is the volume that the data set resides on.

Note: As stated in *z/OS Security Server RACF Security Administrator's Guide*, if a TSO user has executed a non-controlled program during the current session, and then attempts to access a PADS data set, the attempt fails. The TSO user can in some cases temporarily regain a controlled environment by invoking the controlled program through the TSOEXEC command. See "Special consideration when REXX is involved" on page 53 for exceptions to this. When writing a program, you can do the equivalent by invoking the TSO IKJEFTSR service. This technique is useful to users who want to have their programs run from the TSO session, but do not want to protect every program that is executed between logon time and execution of the program intended to access the data set.

Also, program AAOEFTB3 might require protection if the MVS/TSO Dynamic Steplib Facility, program number 5798-DZW, is used while attempting to implement program control in the environment created by TSOEXEC. AAOEFTB3 is normally found in SYS1.LINKLIB.

Trace example 2

There are instances in which the SLIP trace specified in "Trace example 1" on page 50 might not detect all the modules that should be protected. This SLIP should be used with the SLIP trace in "Trace examples" on page 50 if you are still failing with an ABEND 913.

With this trap, it is not necessary to check the value in general register 15. All modules that appear in the trace records need to be protected.

Note: The same module name might appear multiple times in the trace output. Also, you might not see a library name in the trace.

To determine the additional programs that need to be protected use:

```
SLIP SET,IF,ACTION=TRACE,LPAMOD=(ICHRFR00,xxxxx),J=jobname,
TRDATA=(STD,REGS,zzzzz),END
```

Table 16. RMID levels, yyy and zzzzz values for example 2

RMID LEVEL	yyy	zzzzz
UW03221	8A8	9R?+A0?,+7,9R?+A4?,+2B,9R?+A8?,+5
UW08914	8E4	9R?+A0?,+7,9R?+A4?,+2B,9R?+A8?,+5
UW16891	D2	9R?+A0?,+7,9R?+A4?,+2B,9R?+A8?,+5
HRF2220	76	9R?+00?,+7,9R?+04?,+2B,9R?+08?,+5
UW21213	78	9R?+00?,+7,9R?+04?,+2B,9R?+08?,+5
HRF2240 to HRF2608	98	9R?+00?,+7,9R?+04?,+2B,9R?+08?,+5
HRF7703	9C	9R?+00?,+7,9R?+04?,+2B,9R?+08?,+5

Table 16. RMID levels, yyy and zzzzz values for example 2 (continued)		
RMID LEVEL	**yyy**	**zzzzz**
HRF7707	B4	9R?+00?,+7,9R?+04?,+2B,9R?+08?,+5
UA81995 / UA81996 / UA81997	BC	9R?+00?,+7,9R?+04?,+2B,9R?+08?,+5)
UA83891 / UA83892 / UA83893 to current	B4	9R?+00?,+7,9R?+04?,+2B,9R?+08?,+5)

As with the previous TRACE, you need to run an AMBLIST of ICHRFR00 to determine the offset of ICHRCP00. Also, GTF must be started with TRACE=SLIP.

Special consideration when REXX is involved

As mentioned previously, the TCB contains a bit that indicates if the current environment is trusted to open a program accessed data set (PADS) or not. This bit, the TCBNCTL, is turned on by RACF when an unprotected and "untrusted" program is loaded into the environment.

The TCB dirty bit can also be turned on by TSO/E itself. If so, none of the RACF traces detect this. If you issue TSOEXEC pgm_name, which called IRXINIT to create a REXX environment, TSO/E would mark the TCBs on the parallel side "dirty", if the job-step TCB is "dirty". If there are any stack-related commands in the REXX exec, TSO/E checks the job-step TCB and marks the parallel side dirty if the job-step dirty bit is on. Examples of stack-related commands are: PUSH, PULL, QUEUE, MAKEBUF.

Even though TSOEXEC creates a "clean" parallel side in a TSOE address space, TSOE can mark the parallel side "dirty" if the REXX exec is being used and depending on what REXX is doing.

Special consideration when using IMS

DFSDLOC0 is an IMS module that issued an OPEN. This module is not loaded through Contents Supervisor in which case RACF is not called. You cannot use PROGRAM CONTROL/PADS if DFSDLOC0 is involved.

Obtaining traces on z/OS UNIX System Services

When access to a resource is allowed (or denied) incorrectly, you can obtain more data about the problem by using the RACF SET TRACE command, or with GTF trace. GTF trace must be activated to obtain trace output just before reproducing the problem.

To obtain a trace when access is incorrectly denied, using the RACF SET TRACE command. See “Obtaining trace records using the SET TRACE command” on page 55.

To obtain a trace with GTF trace, the trace must be activated to obtain trace output just before reproducing the problem. You may use this procedure:

1. To start GTF, enter this command:

   ```
   START GTF,TRACE=SLIP
   ```

2. Set this SLIP trap for callable services:

   ```
   SLIP SET,IF,LPAEP=(IRRRFR10,0,0),ACTION=TRACE,JOBNAME=xxx,
        TRDATA=(STD,REGS,1R?,+100),END
   ```

   where *xxx* is the job name of a batch job or the user ID of an interactive user.

   This SLIP trap produces a GTF trace entry each time a RACF callable service is invoked (IRRRFR10 is the module called by the RACF callable service). The trace entry contains the parameter list passed with the RACF callable service request.

3. Run the job that has the access problem (or ask the user to attempt to gain access to the resource again) while GTF is on.
4. Stop GTF after the job has ended or after the user has attempted access.
5. Examine the GTF trace output. See *z/OS MVS IPCS Commands* for more information.

Obtaining information about z/OS UNIX file and directory violations

An error occurs when RACF detects an attempt to specify a z/OS UNIX function for which the user does not have authority. When an ICH408I message is issued, it contains a syscall-name that identifies the z/OS UNIX callable service that invoked RACF. In some cases, the message indicates that you do not have sufficient authority to perform the callable service because it requires superuser authority. Superuser authority is UID 0, or authority to an appropriate FACILITY class or UNIXPRIV class profile. In other cases, the message indicates that you do not have access to a file or a directory, and contains your access intent and allowed access.

This message indicates that you do not have permission to find the file pointed to by path name /u/myuser/path:

```
ICH408I USER(MYUSER  ) GROUP(MYGROUP ) NAME(ME)
/u/myuser/path  CL(DIRSRCH ) FID(01C7C3E6E5D4E400011E000000000)
INSUFFICIENT AUTHORITY TO LOOKUP
ACCESS INTENT(--X)  ACCESS ALLOWED(GROUP ---)
EFFECTIVE UID (0000000023) EFFECTIVE GID (0000000012)
```

The final line of the message indicates the UID and GID upon which RACF has based its decision. The first thing to verify is that the UID and GID are the expected values for the failing user ID. In some cases, the values may not correspond to the actual user ID that performed the function (identified within the message). For example, if you execute a set-uid file, which runs under the authority of the file owner, the UID displayed is the file owner. In this case, it is the file owner, and not the end user, who lacks authority to the file or directory.

In the message, a syscall-name of LOOKUP or OPEN, a class name of DIRSRCH, and an access intent of X are all indicators that you do not have authority to a directory in the path name.

This lists the permissions encoded by the file permission bits:

r- Read permission
Are you allowed to read the file/directory?

w - Write permission
Are you allowed to write to the file/directory?

x - Execute permission
Are you allowed to execute the file?

x - Lookup permission
Are you allowed to traverse the directory?

Correcting the permissions when access is denied

To correct the permissions, you should do:

- Verify that every directory that is contained in the path has the execute permission bit set.

  Examine the path that is printed in the message. For every intermediate directory, including the root, issue the command:

  ```
  ls -alLd directory-name
  ```

  If the path is /a/b/c, the commands are:

  ```
  ls -alLd /
  ls -alLd /a
  ```

```
ls -alLd /a/b
ls -alLd /a/b/c
```

Verify that the execute bits are set by examining the output of those commands and correct, if necessary.

```
chmod +x filename
```

Note: After granting this permission, another ICH408I message may be generated, which although similar, is different in the file ID.

- Use the file ID (FID) printed in the message to determine the directory that needs to have permission granted.

  Obtain the auditid tool, which searches the file system for a particular file ID contained in the message. The auditid tool can be found in the z/OS Tools and Toys repository (github.com/IBM/IBM-Z-zOS/tree/main/zOS-Tools-and-Toys).

  List the directory using this command:

```
ls -alLd directory-name
```

  Verify that the execute bits are set by examining the output of this command and correct, if necessary.

```
chmod +x filename
```

  Repeat, if necessary, until there are no more violations.

See *z/OS UNIX System Services Programming: Assembler Callable Services Reference* for more information about syscall functions or *z/OS UNIX System Services Planning* for setting up security for files and directories.

Obtaining trace records using the SET TRACE command

The SET TRACE(APPC | NOAPPC | ASID | NOASID | CALLABLE | NOCALLABLE | CLASS | NOCLASS | ALLCLASSES | IFCLASS | NEVERCLASS | DATABASE | NODATABASE | IMAGE | NOIMAGE | JOBNAME | NOJOBNAME | CALLABLE | NOCALLABLE | RACROUTE | NORACROUTE | SYSTEMSSL | USERID | NOUSERID | ALLUSERIDS | IFUSERID | NEVERUSERID | GENERICANCHOR | NOGENERICANCHOR | RRSF | NORRSF) command specifies whether or not Generalized Trace Facility (GTF) records should be created for the specified events. If the TRACE operand is specified, at least one suboperand is required. The record created is EF44 for each trace event. See *z/OS MVS IPCS User's Guide* or *z/OS MVS Diagnosis: Tools and Service Aids* for information about viewing these records.

Note: These records are intended only for diagnosis use when requested by the IBM support center. The format will not be documented.

Attention: Trace records might contain passwords and therefore, trace output data sets should be appropriately protected.

Command tracing

Command tracing can be useful when diagnosing command errors and can be used with any command in *z/OS Security Server RACF Command Language Reference* that supports the AT keyword. It provides a step-by-step history of how the command text is parsed and rebuilt by the RACF command envelope module and can be used to determine which TSO macros are used. It can help determine if the problem is in the command envelope module or the command processor load module and also determine where a failure occurred during command parsing and rebuilding.

To obtain a command trace, do:

1. At the operator console, activate SET TRACE(IMAGE).
2. From a TSO terminal in TSO READY mode or from ISPF Option 6 (Command), do one or more of:

- Issue a RACF command and append the two characters *-c* (note there's a leading blank between the command and the two characters) to obtain a trace of the command buffer. This keyword is the most useful and is recommended over the others.
- Issue a RACF command and append the two characters *-t* (note there's a leading blank between the command and the two characters) to obtain a trace of the TSO macros used during the parsing of the RACF command and to obtain a trace of the command buffer.
- Issue a RACF command and append the two characters *-n* (note there's a leading blank between the command and the two characters) to not run the command.

3. Output is sent to the user's TSO terminal.

Note:

1. The three characters that are added to the end of the command must be entered exactly as shown and in lowercase letters only. That is, the command image must be followed by one blank space, either *-c*, *-t*, or *-n* and no trailing blanks.
2. Each of these keywords does one thing, but can be issued on the same command together to get multiple results. For example, *-c -n* would trace, but not run the command.
3. If you front-end commands, ensure that this information about the command buffer is passed to RACF:
   - The command alias name (for example, AG, ADDGROUP, and so on) is still the first thing in the buffer
   - The *-c*, *-t*, and *-n* are the last characters in the buffer (with no trailing blanks).
4. Do not use the common command exit, IRREVX01, to put trace characters in the command buffer. Doing this causes a parse error.

See “Front-ending RACF commands” on page 237 for additional information about front-ending RACF commands.

Example of command tracing

Command tracing would be helpful in this example. You entered the command, *ALTUSER MURPHY DATA('PRESIDENT - CATSKILL DIVISION')*, and the command seemed to run, but the installation data field was not updated. By reissuing the command with the *-c* appended to the end of it, *ALTUSER MURPHY DATA('PRESIDENT - CATSKILL DIVISION') -c*, (note the leading blank between the command and the two characters) you can quickly determine if the RACF command envelope module parsed and rebuilt the command image properly before invoking the ALTUSER command processor module.

The command envelope module does multiple parses and rebuilds of the RACF command. If the traced image has the correct installation data, the problem exists in the command processor or some deeper level of RACF. If the traced image does not have the correct installation data, you isolated the problem to the command envelope module.

Figure 1 on page 57 shows the command:

```
ALTUSER MURPHY DATA('PRESIDENT - CATSKILL DIVISION') -c
```

being issued, traced, and rebuilt correctly with all the proper keywords. Therefore, the problem must exist in the command processor or some deeper level of RACF.

```
IRRV099I <ALU>
IRRV099I <ALU>
IRRV099I <ALU>
IRRV099I <ALU       (>
IRRV099I <ALU       (MURPHY>
IRRV099I <ALU       (MURPHY)>
IRRV099I <ALU       (MURPHY)>
IRRV099I <ALU       (MURPHY) DATA>
IRRV099I <ALU       (MURPHY) DATA(>
IRRV099I <ALU       (MURPHY) DATA('>
IRRV099I <ALU       (MURPHY) DATA('PRESIDENT - CATSKILL DIVISION>
IRRV099I <ALU       (MURPHY) DATA('PRESIDENT - CATSKILL DIVISION'>
IRRV099I <ALU       (MURPHY) DATA('PRESIDENT - CATSKILL DIVISION'>
IRRV099I <ALU       (MURPHY) DATA('PRESIDENT - CATSKILL DIVISION' )>
IRRV099I <ALU>
IRRV099I <ALU>
IRRV099I <ALU>
IRRV099I <ALU       (>
IRRV099I <ALU       (MURPHY>
IRRV099I <ALU       (MURPHY)>
IRRV099I <ALU       (MURPHY)>
IRRV099I <ALU       (MURPHY) DATA>
IRRV099I <ALU       (MURPHY) DATA(>
IRRV099I <ALU       (MURPHY) DATA('>
IRRV099I <ALU       (MURPHY) DATA('PRESIDENT - CATSKILL DIVISION>
IRRV099I <ALU       (MURPHY) DATA('PRESIDENT - CATSKILL DIVISION'>
IRRV099I <ALU       (MURPHY) DATA('PRESIDENT - CATSKILL DIVISION'>
IRRV099I <ALU       (MURPHY) DATA('PRESIDENT - CATSKILL DIVISION' )>
```

Figure 1. Trace output showing a problem in the command processor or RACF

Tracing GENERICANCHOR

GENERICANCHOR specifies that tracing is to be in effect for generic anchor loading events. When RACF creates a new generic anchor list for an applicable job, it records a trace record providing the HLQ or class name of the profile list it is creating, the number of profile names in the list, the number of anchors present for the job, and, if applicable, the HLQ or class name for the list that RACF is replacing. Tracing occurs for jobs selected based on the ASID (or ALLASIDS), JOBNAME (or ALLJOBNAMES), or USER ID (or ALLUSERIDS, IFUSERID) keywords.

NOGENERICANCHOR specifies that RACF should not trace generic anchor loading events.

Tracing the R_admin callable service

RACF provides a component trace ability for the R_admin callable service. To activate this trace, start GTF specifying TRACE=USR, and issue the RACF SET TRACE(IMAGE) command. When tracing of IMAGE events is active, the R_admin callable service records a snapshot of the input data it receives in GTF records, including:

- The function-specific parameter list containing segment and field data
- The caller's jobname, ASID, PSW, and ASCB address

It also logs, in a separate record:

- The return code
- The reason code
- The output messages

Tracing the callable services, RACROUTE, and RACF database manager request calls

RACF provides a trace facility that allows tracing RACROUTEs, Callable Services, and RACF Database Manager Requests. When tracing these services, the trace routine will copy the parameter lists into a GTF record before and after the function executes. IPCS is used to view the trace data. To obtain these traces:

1. Start the GTF using the GTFRACF Figure 2 on page 58 or other procedure:

```
START GTFRACF.GTF,,,NOPROMPT
```

2. Use the SET command to enable your trace:

```
@SET TRACE(RACROUTE(TYPE(5)) JOBNAME(IBMUSER))
```

3. Reproduce the scenario that you want. For example, start batch job, log in, start application, use CICS application or access resource.
4. Next stop GTF to prevent excessive traces:

```
STOP GTF
```

5. Use IPCS to view the trace data. The input trace data is contained in the data set specified on the IEFRDER DD card in the GTFRACF (or other) procedure. The sample GTFRACF procedure specifies 'SYS1.TRACE'. Once the TSO IPCS session is active the IPCS subcommand 'IP GTF USR' may be used to display the formatted trace.

Note: The SET TRACE command sets data in the IRRSAFT control block, see Appendix B, “The Security Product Trace Area (SAFT),” on page 261. This information can be used by IBM support to help diagnosis SAF TRACE problems.

Performance Considerations:

- Security as implemented on the OS/390, z/OS platform includes many calls to the security product. This trace facility can adversely affect system performance by adding to the path length associated with performance sensitive security functions. This trace should only be used as a debugging aid. Caution should be exercised when designing the trace (as with any other trace) to impose the least performance penalty. For example, if the address space ID or jobname is known, use these to restrict the scope of the trace.

Usage hints:

- RACF database contention
  - The scope of RACF database serialization depends on the RACF database manager parameter list. Set up a trace for Manager ALTER requests on the specific ASID indicated using GRS contention displays. Examine the trace records to find out what RACF is changing in the database so frequently.
  - When applications specify subsystem and requester information, this is contained in the trace record header. You can determine who is issuing the security function.

```
//GTFRACF PROC MEMBER=GTFPRM#O
//BR14     EXEC PGM=IEFBR14,REGION=512K
//SYSPRINT DD SYSOUT=*
//D DD DISP=(OLD,DELETE),UNIT=3380,VOL=SER=TEMP01,
//  DSN=SYS1.TRACE
//IEFPROC EXEC PGM=AHLGTF,PARM='MODE=EXT,DEBUG=NO,SA=100K,AB=100K',
//             REGION=2880K,TIME=NOLIMIT
//IEFRDER DD   DSNAME=SYS1.TRACE,UNIT=3380,VOL=SER=TEMP01,
//             DISP=(NEW,CATLG),SPACE=(TRK,(100))
//SYSLIB  DD   DSNAME=RACFDRVR.PARMLIB.R6(&MEMBER),DISP=SHR
```

Figure 2. Sample Proc 1: GTFRACF

```
TRACE=USRP
USR=(F44),END
```

Figure 3. Sample parmlib member: GTFPRM#O

Reading a trace output

The trace output formatted by IPCS is split into 3 main areas. The first area contains common information for all services. For example, the caller's information and return codes. The parameter lists and unloaded

parameters are found in the second area. The third area is a complete hex dump of the entire GTF record which includes the header information, parameter lists and all of the parameters that were unloaded.

For RACROUTE requests, there are two parameters lists. The first one is the SAF parameter list mapped by ICHSAFP. The second parameter list is mapped by the specific RACROUTE type parameter list. These are described in *z/OS Security Server RACF Data Areas* in the z/OS Internet library (www.ibm.com/servers/resourcelink/svc00100.nsf/pages/zosInternetLibrary). The parameter lists for Callable Services can be found in *z/OS Security Server RACF Callable Services*. For Manager calls, see the Diagnosis Guide for the Manager parameter list mapping.

Following the parameter lists are the parameters that are unloaded. Not all parameters are unloaded. Before most parameters are unloaded, there are extra entries that say OFFSET##. This is for informational purposes to let you know that the following parameter is at offset ## in the parameter list. This helps determine what parameter you are examining.

Due to nesting of some services PRE and POST trace records might not be in sequential order. For example, one might see two PRE calls and then two POST calls.

Explaining trace header information

Trace identifier
SAFTRACE Record type 54.

Record eyecatcher
Eyecatcher for the SAFTRACE record type: RTRACE.

Trace type
Identifies what trace this is. Types are OMVS for callable services, MNGR for manager calls, or RACF for RACROUTE traces. These types are followed by PRE or POST to indicate that the record pertains to a pre-function trace or post-function trace.

Ending sequence
This contains a unique identifier in the case of the trace record exceeded the maximum size. If an there is an abend during the trace, it contains RECOVERY. Ending sequences indicate that a problem occurred during trace capture. If you find these, it is indicative of a problem with the parameter list built by the caller of the function.

Calling address
Address of the caller. This is useful along with a dump of the address space when determining the caller of the service in question. It may not be possible to determine the caller of the function using the Request/Subsystem (when available).

Requestor/Subsystem
Caller's SAFPREQR and SAFPSUBS as specified on RACROUTE.

Primary jobname
Primary jobname.

Primary asid
Primary ASID.

Primary ACEEP
Primary ASID related ACEE pointer, as indicated in ASXBSENV.

Home jobname
Home Jobname.

Home asid
Home ASID.

Home ACEEP
Home ASID related ACEE pointer as indicated in the home address space ASXBSENV field.

Task address
TCB pointer of current task.

Time
Time stamp.

Error class
Possible Security product resource class name in error.

Service number
Service number for Callable service or RACROUTE. (See Table 17 on page 60 and Table 18 on page 62 for supported services).

RACF Return code
RACF Return Code.

RACF Reason code
RACF Reason Code.

Return area address
Address of data returned by the function. When used with a dump of an address space, storage creep problems can be isolated to the application.

Parameter count
Number of parameters that were captured. Note, this is not the number of parameters in the parmlist, but rather the number of parameters which were collected.

Tables

Table 17. Callable services type numbers

CALLABLE SERVICE	Service Number or TYPE (HEX)	Service Number or TYPE (DECIMAL)
IRRSIU00	1	1
IRRSDU00	2	2
IRRSMF00	3	3
Reserved	4	4
IRRSMM00	5	5
IRRSKA00	6	6
IRRSKP00	7	7
IRRSUM00	8	8
IRRSGM00	9	9
IRRSGG00	A	10
IRRSSU00	B	11
IRRSEU00	C	12
IRRSSG00	D	13
IRRSEG00	E	14
IRRSCO00	F	15
IRRSCF00	10	16
IRRSCA00	11	17
IRRSEX00	12	18
IRRSAU00	13	19
IRRSKO00	14	20
IRRSQS00	15	21

Table 17. Callable services type numbers (continued)

CALLABLE SERVICE	Service Number or TYPE (HEX)	Service Number or TYPE (DECIMAL)
IRRSQF00	16	22
IRRSCS00	17	23
IRRSKF00	18	24
IRRSMR00	19	25
IRRSPT00	1A	26
IRRSUG00	1B	27
IRRSFK00	1C	28
IRRSM100	1D	29
IRRSK100	1E	30
IRRSC100	1F	31
IRRSC200	20	32
IRRSGE00	21	33
IRRSD100	22	34
IRRSDK00	23	35
IRRSUD00	24	36
IRRSDA00	25	37
IRRSIA00	26	38
IRRSEQ00*	27	39
IRRSIM00	28	40
IRRSDL00	29	41
IRRSMK00	2A	42
IRRSPK00	2B	43
IRRSPX00	2C	44
IRRSCH00	2D	45
IRRSPY00	2E	46
IRRSCL00	2F	47
IRRSSB00	30	48
IRRSWP00	31	49
IRRSGS00	32	50
IRRSAX00	33	51
IRRSGI00	34	52
IRRSPS00	35	53
IRRSPW00	36	54

Note: Callable Service IRRSEQ00, R_Admin, has its own trace facility.

Table 18. RACROUTE CALL= service type number

RACROUTE CALL=	Service / Type Number in Hex	Service / Type Number in Decimal
AUTH	1	1
FASTAUTH	2	2
LIST	3	3
DEFINE	4	4
VERIFY	5	5
EXTRACT	6	6
DIRAUTH	7	7
TOKENMAP	8	8
VERIFYX	9	9
TOKENXTR	A	10
TOKENBLD	B	11
EXTRACT, BR=YES	C	12
AUDIT	D	13
STAT	E	14
SIGNON	F	15
TOKENMAP, XMEM	10	16
TOKENXTR, XMEM	11	17

Examples of trace output

When an ACEE is dumped in the trace record, the associated ACEX, USP, and TOKEN are also dumped with it.

Callable Service

This is an example of SAF trace output for the R_dceauth Callable Service (service type 37) PRE and POST entries.

```
  **** GTFTRACE DISPLAY OPTIONS IN EFFECT ****
USR=ALL

**** GTF DATA COLLECTION OPTIONS IN EFFECT: ****
USRP option


         **** GTF TRACING ENVIRONMENT ****
     Release: SP7.0.3   FMID: HBB7706   System name: RACFR13
     CPU Model: 4381  Version: FF  Serial no. 111515


USR57 F44 ASCB 00FA9500           JOBN J23DC002


    Following is a formatted R_TRACE record.
     This trace record was generated by IRRTRC00 with IDENT(R_TRACE).

     Trace Identifier:            00000036
     Record Eyecatcher:           RTRACE
     Trace Type:                  OMVSPRE
     Ending Sequence:             ........
```

```
Calling address:              00000000  8B000BAE
Requestor/Subsystem:          ........  ........
Primary jobname:              J23DC002
Primary asid:                 0000012E
Primary ACEEP:                00000000  006F6700
Home jobname:                 J23DC002
Home asid:                    0000012E
Home ACEEP:                   00000000  006F6700
Task address:                 00000000  006F6B00
Task ACEEP:                   00000000  00000000
Time:                         B570BA31  042B3327
Error class:                  ........
Service number:               00000025
RACF Return code:             00000000
RACF Reason code:             00000000
Return area address:          00000000  00000000
Parameter count:              00000025


Area length:                  00000040

Area value:
0B001B2C  0B0016EC  0B0016F0  0B0016EC  | ..........0.... |
0B0016F4  0B0016EC  0B0016F8  0B001700  | ...4.......8.... |
0B0016EC  0B0019A1  0B0019C5  0B0019E9  | .......~...E...Z |
0B0019F2  0B001AE8  0B001704  0B001706  | ...2...Y........ |

Area length:                  00000008

Area value:
D6C6C6E2  C5E30004                    | OFFSET..         |

Area length:                  00000004

Area value:
00000000                              | ....             |

Area length:                  00000008

Area value:
D6C6C6E2  C5E30008                    | OFFSET..         |

Area length:                  00000004

Area value:
00000000                              | ....             |

Area length:                  00000008

Area value:
D6C6C6E2  C5E3000C                    | OFFSET..         |

Area length:                  00000004

Area value:
00000000                              | ....             |

Area length:                  00000008

Area value:
D6C6C6E2  C5E30010                    | OFFSET..         |

Area length:                  00000004

Area value:
00000000                              | ....             |

Area length:                  00000008

Area value:
D6C6C6E2  C5E30014                    | OFFSET..         |

Area length:                  00000004

Area value:
00000000                              | ....             |

Area length:                  00000008

Area value:
D6C6C6E2  C5E30018                    | OFFSET..         |
```

```
Area length:                  00000004

Area value:
00000000                                | ....            |

Area length:                  00000008

Area value:
D6C6C6E2  C5E3001C                      | OFFSET..        |

Area length:                  00000004

Area value:
7FFF5268                                | "...            |

Area length:                  000000A8

Area value:
C1C3C5C5  FF0000A8  02000000  00000000  | ACEE...y........ |
00000000  05C8D6D5  C4C14040  4004E2E8  | .....HONDA   .SY |
E2F14040  40400101  0001054F  40404040  | S1    .....|     |
40404040  00000000  00000000  00000000  |     ............ |
40404040  40404040  00000000  00000000  |         ........ |
00000000  00000000  40404040  40404040  | ........        |
00000000  7FFD3258  00000000  0101054F  | ....".........| |
00000000  00A00000  00000000  00000000  | ................ |
00000000  00000000  7FFD3290  00000000  | ........".....  |
00000000  7FFD3968                      | ....".          |

Area length:                  00000050

Area value:
50010000  0000C000  00000000  00000000  | &.....{......... |
00000000  00000000  00000000  00000000  | ................ |
00000000  00000000  00000000  00000000  | ................ |
00000000  00000000  00000000  00000000  | ................ |
C8D6D5C4  C1404040  E2E8E2F1  40404040  | HONDA   SYS1     |

Area length:                  00000090

Area value:
C1C3C5E7  03000000  00000000  00000000  | ACEX............ |
00000000  00000000  00000000  00000000  | ................ |
00000000  00000024  7FFD3000  00000000  | ........".....  |
00000000  00000000  00000000  00000000  | ................ |
00000000  00000000  00000000  00000000  | ................ |
00000000  00000000  00000000  00000000  | ................ |
00000000  00000000  00000000  00000000  | ................ |
00000000  00000000  00000000  00000000  | ................ |
00000000  00000000  00000000  00000000  | ................ |

Area length:                  00000008

Area value:
D6C6C6E2  C5E30020                      | OFFSET..        |

Area length:                  00000004

Area value:
00000000                                | ....            |

Area length:                  00000008

Area value:
D6C6C6E2  C5E30024                      | OFFSET..        |

Area length:                  00000024

Area value:
81828384  85868182  60838485  86608182  | abcdefab-cdef-ab |
838460F1  F1818160  F4F5F4F5  F4F5F4F5  | cd-11aa-45454545 |
F4F5F4F5                                | 4545            |

Area length:                  00000008

Area value:
D6C6C6E2  C5E30028                      | OFFSET..        |

Area length:                  00000024

Area value:
81828384  85868182  60838485  8660F2F3  | abcdefab-cdef-23 |
```

```
F3F260F1  F1818160  F4F5F4F5  F4F5F4F5  | 32-11aa-45454545 |
F4F5F4F5                                | 4545             |

Area length:                  00000008

Area value:
D6C6C6E2  C5E3002C                      | OFFSET..         |

Area length:                  00000006

Area value:
05C8D6D5  C4C1                          | .HONDA           |

Area length:                  00000008

Area value:
D6C6C6E2  C5E30030                      | OFFSET..         |

Area length:                  00000008

Area value:
C6C1C3C9  D3C9E3E8                      | FACILITY         |

Area length:                  00000008

Area value:
D6C6C6E2  C5E30034                      | OFFSET..         |

Area length:                  00000006

Area value:
D7D9D6E3  E6D6                          | PROTWO           |

Area length:                  00000008

Area value:
D6C6C6E2  C5E30038                      | OFFSET..         |

Area length:                  00000002

Area value:
0006                                    | ..               |

Area length:                  00000008

Area value:
D6C6C6E2  C5E3003C                      | OFFSET..         |

Area length:                  00000001

Area value:
04                                      | .                |

Area length:                  000000A8

Area value:
C1C3C5C5  FF0000A8  02000000  00000000  | ACEE...y........ |
00000000  07C9C2D4  E4E2C5D9  4004E2E8  | .....IBMUSER .SY |
E2F14040  4040A110  0001054F  40404040  | S1    ~....|     |
40404040  00000000  00000000  00000000  |     ............ |
40404040  40404040  00000000  00000000  |         ........ |
00000000  00000000  40404040  40404040  | ........         |
00000000  006F67A8  00000000  00000000  | .....?.y........ |
00000000  006F67C0  00000000  0101054F  | .....?.{.......| |
00000000  00200000  00000000  00000000  | ................ |
00000000  006F6888                      | .....?.h         |

Area length:                  00000050

Area value:
50010207  0003C000  00000000  00000000  | &.....{......... |
D7D6D2E5  D4D4C3D3  C9C2D4E4  E2C5D940  | POKVMMCLIBMUSER  |
D7D6D2E5  D4D4C3D3  E2E8E2F1  40404040  | POKVMMCLSYS1     |
C9D5E3D9  C4D94040  00000000  00000000  | INTRDR  ........ |
C9C2D4E4  E2C5D940  E2E8E2F1  40404040  | IBMUSER SYS1     |

Area length:                  00000090

Area value:
C1C3C5E7  03000000  00FAB6F8  00000000  | ACEX.......8.... |
00000000  00000000  00000000  00000000  | ................ |
00000000  00000024  006F68D8  00000000  | .........?.Q.... |
```

```
      00000000  00000000  00000000  00000000  | ................ |
      00000000  00000000  00000000  00000000  | ................ |
      00000000  00000000  00000000  00000000  | ................ |
      00000000  00000000  00000000  00000000  | ................ |
      00000000  00000000  00000000  00000000  | ................ |
      00000000  00000000  00000000  00000000  | ................ |

      Hexadecimal dump of record follows:
    +0000  00000036  D9E3D9C1  C3C54040  D6D4E5E2  | ....RTRACE  OMVS |
    +0010  D7D9C540  00000000  00000000  00000000  | PRE ............ |
    +0020  00000000  00000000  00000000  8B000BAE  | ................ |
    +0030  00000000  00000000  00000000  00000000  | ................ |
    +0040  00000000  00000000  006F6B00  00FA9500  | .........?,...n. |
    +0050  00FA9500  0000012E  0000012E  D1F2F3C4  | ..n.........J23D |
    +0060  C3F0F0F2  D1F2F3C4  C3F0F0F2  006F6700  | C002J23DC002.?.. |
    +0070  006F6700  00000000  B570BA31  042B3327  | .?.............. |
    +0080  00000000  00000000  00000025  00000025  | ................ |
    +0090  00000040  0B001B2C  0B0016EC  0B0016F0  | ... ...........0 |
    +00A0  0B0016EC  0B0016F4  0B0016EC  0B0016F8  | .......4.......8 |
    +00B0  0B001700  0B0016EC  0B0019A1  0B0019C5  | ..........~...E |
    +00C0  0B0019E9  0B0019F2  0B001AE8  0B001704  | ...Z...2...Y.... |
    +00D0  0B001706  00000008  D6C6C6E2  C5E30004  | ........OFFSET.. |
    +00E0  00000004  00000000  00000008  D6C6C6E2  | ............OFFS |
    +00F0  C5E30008  00000004  00000000  00000008  | ET.............. |
    +0100  D6C6C6E2  C5E3000C  00000004  00000000  | OFFSET.......... |
    +0110  00000008  D6C6C6E2  C5E30010  00000004  | ....OFFSET...... |
    +0120  00000000  00000008  D6C6C6E2  C5E30014  | ........OFFSET.. |
    +0130  00000004  00000000  00000008  D6C6C6E2  | ............OFFS |
    +0140  C5E30018  00000004  00000000  00000008  | ET.............. |
    +0150  D6C6C6E2  C5E3001C  00000004  7FFF5268  | OFFSET......"... |
    +0160  000000A8  C1C3C5C5  FF0000A8  02000000  | ...yACEE...y.... |
    +0170  00000000  00000000  05C8D6D5  C4C14040  | .........HONDA   |
    +0180  4004E2E8  E2F14040  40400101  0001054F  |  .SYS1    .....| |
    +0190  40404040  40404040  00000000  00000000  |          ........ |
    +01A0  00000000  40404040  40404040  00000000  | ....         .... |
    +01B0  00000000  00000000  00000000  40404040  | ............     |
    +01C0  40404040  00000000  7FFF5000  00000000  |     ....".&..... |
    +01D0  00000000  00000000  7FFD3258  00000000  | ........".2..... |
    +01E0  0101054F  00000000  00A00000  00000000  | ...|............ |
    +01F0  00000000  00000000  00000000  7FFD3290  | ............".2. |
    +0200  00000000  00000000  7FFD3968  00000050  | ........".2...&  |
    +0210  50010000  0000C000  00000000  00000000  | &.....{......... |
    +0220  00000000  00000000  00000000  00000000  | ................ |
    +0230  00000000  00000000  00000000  00000000  | ................ |
    +0240  00000000  00000000  00000000  00000000  | ................ |
    +0250  C8D6D5C4  C1404040  E2E8E2F1  40404040  | HONDA   SYS1     |
    +0260  00000090  C1C3C5E7  03000000  00000000  | ....ACEX........ |
    +0270  00000000  00000000  00000000  00000000  | ................ |
    +0280  00000000  00000000  00000024  7FFD3000  | ............".0. |
    +0290  00000000  00000000  00000000  00000000  | ................ |
    +02A0  00000000  00000000  00000000  00000000  | ................ |
    +02B0  00000000  00000000  00000000  00000000  | ................ |
    +02C0  00000000  00000000  00000000  00000000  | ................ |
    +02D0  00000000  00000000  00000000  00000000  | ................ |
    +02E0  00000000  00000000  00000000  00000000  | ................ |
    +02F0  00000000  00000008  D6C6C6E2  C5E30020  | ........OFFSET.. |
    +0300  00000004  00000000  00000008  D6C6C6E2  | ............OFFS |
    +0310  C5E30024  00000024  81828384  85868182  | ET......abcdefab |
    +0320  60838485  86608182  838460F1  F1818160  | -cdef-abcd-11aa- |
    +0330  F4F5F4F5  F4F5F4F5  F4F5F4F5  00000008  | 454545454545.... |
    +0340  D6C6C6E2  C5E30028  00000024  81828384  | OFFSET......abcd |
    +0350  85868182  60838485  8660F2F3  F3F260F1  | efab-cdef-2332-1 |
    +0360  F1818160  F4F5F4F5  F4F5F4F5  F4F5F4F5  | 1aa-454545454545 |
    +0370  00000008  D6C6C6E2  C5E3002C  00000006  | ....OFFSET...... |
    +0380  05C8D6D5  C4C10000  0008D6C6  C6E2C5E3  | .HONDA....OFFSET |
    +0390  00300000  0008C6C1  C3C9D3C9  E3E80000  | ......FACILITY.. |
    +03A0  0008D6C6  C6E2C5E3  00340000  0006D7D9  | ..OFFSET......PR |
    +03B0  D6E3E6D6  00000008  D6C6C6E2  C5E30038  | OTWO....OFFSET.. |
    +03C0  00000002  00060000  0008D6C6  C6E2C5E3  | ..........OFFSET |
    +03D0  003C0000  00010400  0000A8C1  C3C5C5FF  | ..........yACEE. |
    +03E0  0000A802  00000000  00000000  00000007  | ..y............. |
    +03F0  C9C2D4E4  E2C5D940  04E2E8E2  F1404040  | IBMUSER .SYS1    |
    +0400  40A11000  01054F40  40404040  40404000  |  ~....|        . |
    +0410  00000000  00000000  00000040  40404040  | ...........      |
    +0420  40404000  00000000  00000000  00000000  |    ............. |
    +0430  00000040  40404040  40404000  00000000  | ...         ..... |
    +0440  6F67A800  00000000  00000000  00000000  | ?.y............. |
    +0450  6F67C000  00000001  01054F00  00000000  | ?.{.......|..... |
    +0460  20000000  00000000  00000000  00000000  | ................ |
    +0470  00000000  6F67F800  00000000  00000000  | ....?.8......... |
    +0480  6F688800  00005050  01020700  03C00000  | ?.h...&&.....{.. |
    +0490  00000000  000000D7  D6D2E5D4  D4C3D3C9  | .......POKVMMCLI |
```

```
  +04A0  C2D4E4E2  C5D940D7  D6D2E5D4  D4C3D3E2  | BMUSER POKVMMCLS |
  +04B0  E8E2F140  404040C9  D5E3D9C4  D9404000  | YS1    INTRDR  . |
  +04C0  00000000  000000C9  C2D4E4E2  C5D940E2  | .......IBMUSER S |
  +04D0  E8E2F140  40404000  000090C1  C3C5E703  | YS1    ....ACEX. |
  +04E0  00000000  FAB6F800  00000000  00000000  | ......8......... |
  +04F0  00000000  00000000  00000000  00000000  | ................ |
  +0500  00002400  6F68D800  00000000  00000000  | ....?.Q......... |
  +0510  00000000  00000000  00000000  00000000  | ................ |
  +0520  00000000  00000000  00000000  00000000  | ................ |
  +0530  00000000  00000000  00000000  00000000  | ................ |
  +0540  00000000  00000000  00000000  00000000  | ................ |
  +0550  00000000  00000000  00000000  00000000  | ................ |
  +0560  00000000  00000000  000000              | ..........       |
               GMT-02/23/2001 15:50:35.825866  LOC-02/23/2001 11:50:35.825866

USR57 F44 ASCB 00FA9500           JOBN J23DC002


    Following is a formatted R_TRACE record.
     This trace record was generated by IRRTRC00 with IDENT(R_TRACE).

    Trace Identifier:             00000036
    Record Eyecatcher:            RTRACE
    Trace Type:                   OMVSPOST
    Ending Sequence:              ........
    Calling address:              00000000  8B000BAE
    Requestor/Subsystem:          ........  ........
    Primary jobname:              J23DC002
    Primary asid:                 0000012E
    Primary ACEEP:                00000000  006F6700
    Home jobname:                 J23DC002
    Home asid:                    0000012E
    Home ACEEP:                   00000000  006F6700
    Task address:                 00000000  006F6B00
    Task ACEEP:                   00000000  00000000
    Time:                         B570BA31  24A9A94B
    Error class:                  ........
    Service number:               00000025
    RACF Return code:             00000000
    RACF Reason code:             00000000
    Return area address:          00000000  00000000
    Parameter count:              00000025


    Area length:                  00000040

    Area value:
    0B001B2C  0B0016EC  0B0016F0  0B0016EC  | ...........0.... |
    0B0016F4  0B0016EC  0B0016F8  0B001700  | ...4.......8.... |
    0B0016EC  0B0019A1  0B0019C5  0B0019E9  | .......~...E...Z |
    0B0019F2  0B001AE8  0B001704  0B001706  | ...2...Y........ |

    Area length:                  00000008

    Area value:
    D6C6C6E2  C5E30004                    | OFFSET..         |

    Area length:                  00000004

    Area value:
    00000000                              | ....             |

    Area length:                  00000008

    Area value:
    D6C6C6E2  C5E30008                    | OFFSET..         |

    Area length:                  00000004

    Area value:
    00000000                              | ....             |

    Area length:                  00000008

    Area value:
    D6C6C6E2  C5E3000C                    | OFFSET..         |

    Area length:                  00000004

    Area value:
    00000000                              | ....             |
```

```
Area length:                  00000008

Area value:
D6C6C6E2  C5E30010                    | OFFSET..         |

Area length:                  00000004

Area value:
00000000                              | ....             |

Area length:                  00000008

Area value:
D6C6C6E2  C5E30014                    | OFFSET..         |

Area length:                  00000004

Area value:
00000000                              | ....             |

Area length:                  00000008

Area value:
D6C6C6E2  C5E30018                    | OFFSET..         |

Area length:                  00000004

Area value:
00000000                              | ....             |

Area length:                  00000008

Area value:
D6C6C6E2  C5E3001C                    | OFFSET..         |

Area length:                  00000004

Area value:
7FFF5268                              | "...             |

Area length:                  000000A8

Area value:
C1C3C5C5  FF0000A8  02000000  00000000  | ACEE...y........ |
00000000  05C8D6D5  C4C14040  4004E2E8  | .....HONDA   .SY |
E2F14040  40400101  0001054F  40404040  | S1    .....|     |
40404040  00000000  00000000  00000000  |     ............ |
40404040  40404040  00000000  00000000  |         ........ |
00000000  00000000  40404040  40404040  | ........         |
00000000  7FFF5000  00000000  00000000  | ....".&......... |
00000000  7FFD3258  00000000  0101054F  | ....".........| |
00000000  00A00000  00000000  00000000  | ................ |
00000000  00000000  7FFD3290  00000000  | ........"....... |
00000000  7FFD3968                      | ....".."         |

Area length:                  00000050

Area value:
50010000  0000C000  00000000  00000000  | &.....{......... |
00000000  00000000  00000000  00000000  | ................ |
00000000  00000000  00000000  00000000  | ................ |
00000000  00000000  00000000  00000000  | ................ |
C8D6D5C4  C1404040  E2E8E2F1  40404040  | HONDA   SYS1     |

Area length:                  00000090

Area value:
C1C3C5E7  03000000  00000000  00000000  | ACEX............ |
00000000  00000000  00000000  00000000  | ................ |
00000000  00000024  7FFD3000  00000000  | ........"....... |
00000000  00000000  00000000  00000000  | ................ |
00000000  00000000  00000000  00000000  | ................ |
00000000  00000000  00000000  00000000  | ................ |
00000000  00000000  00000000  00000000  | ................ |
00000000  00000000  00000000  00000000  | ................ |
00000000  00000000  00000000  00000000  | ................ |

Area length:                  00000008

Area value:
D6C6C6E2  C5E30020                    | OFFSET..         |
```

```
Area length:                  00000004

Area value:
00000000                                | ....              |

Area length:                  00000008

Area value:
D6C6C6E2  C5E30024                      | OFFSET..          |

Area length:                  00000024

Area value:
81828384  85868182  60838485  86608182  | abcdefab-cdef-ab |
838460F1  F1818160  F4F5F4F5  F4F5F4F5  | cd-11aa-45454545 |
F4F5F4F5                                | 4545              |

Area length:                  00000008

Area value:
D6C6C6E2  C5E30028                      | OFFSET..          |

Area length:                  00000024

Area value:
81828384  85868182  60838485  8660F2F3  | abcdefab-cdef-23 |
F3F260F1  F1818160  F4F5F4F5  F4F5F4F5  | 32-11aa-45454545 |
F4F5F4F5                                | 4545              |

Area length:                  00000008

Area value:
D6C6C6E2  C5E3002C                      | OFFSET..          |

Area length:                  00000006

Area value:
05C8D6D5  C4C1                          | .HONDA            |

Area length:                  00000008

Area value:
D6C6C6E2  C5E30030                      | OFFSET..          |

Area length:                  00000008

Area value:
C6C1C3C9  D3C9E3E8                      | FACILITY          |

Area length:                  00000008

Area value:
D6C6C6E2  C5E30034                      | OFFSET..          |

Area length:                  00000006

Area value:
D7D9D6E3  E6D6                          | PROTWO            |

Area length:                  00000008

Area value:
D6C6C6E2  C5E30038                      | OFFSET..          |

Area length:                  00000002

Area value:
0006                                    | ..                |

Area length:                  00000008

Area value:
D6C6C6E2  C5E3003C                      | OFFSET..          |

Area length:                  00000001

Area value:
04                                      | .                 |

Area length:                  000000A8

Area value:
```

```
      C1C3C5C5  FF0000A8  02000000  00000000  | ACEE...y........ |
      00000000  07C9C2D4  E4E2C5D9  4004E2E8  | .....IBMUSER .SY |
      E2F14040  4040A110  0001054F  40404040  | S1    ~....|     |
      40404040  00000000  00000000  00000000  |     ............ |
      40404040  40404040  00000000  00000000  |         ........ |
      00000000  00000000  40404040  40404040  | ........         |
      00000000  006F67A8  00000000  00000000  | .....?.y........ |
      00000000  006F67C0  00000000  0101054F  | .....?.{.......| |
      00000000  00200000  00000000  00000000  | ................ |
      00000000  00000000  006F67F8  00000000  | .........?.8.... |
      00000000  006F6888                      | .....?.h         |

      Area length:                  00000050

      Area value:
      50010207  0003C000  00000000  00000000  | &.....{......... |
      D7D6D2E5  D4D4C3D3  C9C2D4E4  E2C5D940  | POKVMMCLIBMUSER  |
      D7D6D2E5  D4D4C3D3  E2E8E2F1  40404040  | POKVMMCLSYS1     |
      C9D5E3D9  C4D94040  00000000  00000000  | INTRDR  ........ |
      C9C2D4E4  E2C5D940  E2E8E2F1  40404040  | IBMUSER SYS1     |

      Area length:                  00000090

      Area value:
      C1C3C5E7  03000000  00FAB6F8  00000000  | ACEX.......8.... |
      00000000  00000000  00000000  00000000  | ................ |
      00000000  00000024  006F68D8  00000000  | .........?.Q.... |
      00000000  00000000  00000000  00000000  | ................ |
      00000000  00000000  00000000  00000000  | ................ |
      00000000  00000000  00000000  00000000  | ................ |
      00000000  00000000  00000000  00000000  | ................ |
      00000000  00000000  00000000  00000000  | ................ |
      00000000  00000000  00000000  00000000  | ................ |

      Hexadecimal dump of record follows:
    +0000  00000036  D9E3D9C1  C3C54040  D6D4E5E2  | ....RTRACE  OMVS |
    +0010  D7D6E2E3  00000000  00000000  00000000  | POST............ |
    +0020  00000000  00000000  00000000  8B000BAE  | ................ |
    +0030  00000000  00000000  00000000  00000000  | ................ |
    +0040  00000000  00000000  006F6B00  00FA9500  | .........?,...n. |
    +0050  00FA9500  0000012E  0000012E  D1F2F3C4  | ..n.........J23D |
    +0060  C3F0F0F2  D1F2F3C4  C3F0F0F2  006F6700  | C002J23DC002.?.. |
    +0070  006F6700  00000000  B570BA31  24A9A94B  | .?..........zz. |
    +0080  00000000  00000000  00000025  00000025  | ................ |
    +0090  00000040  0B001B2C  0B0016EC  0B0016F0  | ... ...........0 |
    +00A0  0B0016EC  0B0016F4  0B0016EC  0B0016F8  | .......4.......8 |
    +00B0  0B001700  0B0016EC  0B0019A1  0B0019C5  | ...........~...E |
    +00C0  0B0019E9  0B0019F2  0B001AE8  0B001704  | ...Z...2...Y.... |
    +00D0  0B001706  00000008  D6C6C6E2  C5E30004  | ........OFFSET.. |
    +00E0  00000004  00000000  00000008  D6C6C6E2  | ............OFFS |
    +00F0  C5E30008  00000004  00000000  00000008  | ET.............. |
    +0100  D6C6C6E2  C5E3000C  00000004  00000000  | OFFSET.......... |
    +0110  00000008  D6C6C6E2  C5E30010  00000004  | ....OFFSET...... |
    +0120  00000000  00000008  D6C6C6E2  C5E30014  | ........OFFSET.. |
    +0130  00000004  00000000  00000008  D6C6C6E2  | ............OFFS |
    +0140  C5E30018  00000004  00000000  00000008  | ET.............. |
    +0150  D6C6C6E2  C5E3001C  00000004  7FFF5268  | OFFSET......"... |
    +0160  000000A8  C1C3C5C5  FF0000A8  02000000  | ...yACEE...y.... |
    +0170  00000000  00000000  05C8D6D5  C4C14040  | .........HONDA   |
    +0180  4004E2E8  E2F14040  40400101  0001054F  |  .SYS1    .....| |
    +0190  40404040  40404040  00000000  00000000  |          ........ |
    +01A0  00000000  40404040  40404040  00000000  | ....        .... |
    +01B0  00000000  00000000  00000000  40404040  | ............     |
    +01C0  40404040  00000000  7FFF5000  00000000  |     ....".&..... |
    +01D0  00000000  00000000  7FFD3258  00000000  | ........".......  |
    +01E0  0101054F  00000000  00A00000  00000000  | ...|............ |
    +01F0  00000000  00000000  00000000  7FFD3290  | ............"... |
    +0200  00000000  00000000  7FFD3968  00000050  | ........".....& |
    +0210  50010000  0000C000  00000000  00000000  | &.....{......... |
    +0220  00000000  00000000  00000000  00000000  | ................ |
    +0230  00000000  00000000  00000000  00000000  | ................ |
    +0240  00000000  00000000  00000000  00000000  | ................ |
    +0250  C8D6D5C4  C1404040  E2E8E2F1  40404040  | HONDA   SYS1     |
    +0260  00000090  C1C3C5E7  03000000  00000000  | ....ACEX........ |
    +0270  00000000  00000000  00000000  00000000  | ................ |
    +0280  00000000  00000000  00000024  7FFD3000  | ............"... |
    +0290  00000000  00000000  00000000  00000000  | ................ |
    +02A0  00000000  00000000  00000000  00000000  | ................ |
    +02B0  00000000  00000000  00000000  00000000  | ................ |
    +02C0  00000000  00000000  00000000  00000000  | ................ |
    +02D0  00000000  00000000  00000000  00000000  | ................ |
    +02E0  00000000  00000000  00000000  00000000  | ................ |
```

```
  +02F0  00000000  00000008  D6C6C6E2  C5E30020  | ........OFFSET.. |
  +0300  00000004  00000000  00000008  D6C6C6E2  | ............OFFS |
  +0310  C5E30024  00000024  81828384  85868182  | ET......abcdefab |
  +0320  60838485  86608182  838460F1  F1818160  | -cdef-abcd-11aa- |
  +0330  F4F5F4F5  F4F5F4F5  F4F5F4F5  00000008  | 454545454545.... |
  +0340  D6C6C6E2  C5E30028  00000024  81828384  | OFFSET......abcd |
  +0350  85868182  60838485  8660F2F3  F3F260F1  | efab-cdef-2332-1 |
  +0360  F1818160  F4F5F4F5  F4F5F4F5  F4F5F4F5  | 1aa-454545454545 |
  +0370  00000008  D6C6C6E2  C5E3002C  00000006  | ....OFFSET...... |
  +0380  05C8D6D5  C4C10000  0008D6C6  C6E2C5E3  | .HONDA....OFFSET |
  +0390  00300000  0008C6C1  C3C9D3C9  E3E80000  | ......FACILITY.. |
  +03A0  0008D6C6  C6E2C5E3  00340000  0006D7D9  | ..OFFSET......PR |
  +03B0  D6E3E6D6  00000008  D6C6C6E2  C5E30038  | OTWO....OFFSET.. |
  +03C0  00000002  00060000  0008D6C6  C6E2C5E3  | ..........OFFSET |
  +03D0  003C0000  00010400  0000A8C1  C3C5C5FF  | ..........yACEE. |
  +03E0  0000A802  00000000  00000000  00000007  | ..y............. |
  +03F0  C9C2D4E4  E2C5D940  04E2E8E2  F1404040  | IBMUSER .SYS1    |
  +0400  40A11000  01054F40  40404040  40404000  |  ~....|        . |
  +0410  00000000  00000000  00000040  40404040  | ...........      |
  +0420  40404000  00000000  00000000  00000000  |    ............. |
  +0430  00000040  40404040  40404000  00000000  | ...        ..... |
  +0440  6F67A800  00000000  00000000  00000000  | ?.y............. |
  +0450  6F67C000  00000001  01054F00  00000000  | ?.{.......|..... |
  +0460  20000000  00000000  00000000  00000000  | ................ |
  +0470  00000000  6F67F800  00000000  00000000  | ....?.8......... |
  +0480  6F688800  00005050  01020700  03C00000  | ?.h...&&.....{.. |
  +0490  00000000  000000D7  D6D2E5D4  D4C3D3C9  | .......POKVMMCLI |
  +04A0  C2D4E4E2  C5D940D7  D6D2E5D4  D4C3D3E2  | BMUSER POKVMMCLS |
  +04B0  E8E2F140  404040C9  D5E3D9C4  D9404000  | YS1    INTRDR  . |
  +04C0  00000000  000000C9  C2D4E4E2  C5D940E2  | .......IBMUSER S |
  +04D0  E8E2F140  40404000  000090C1  C3C5E703  | YS1    ....ACEX. |
  +04E0  00000000  FAB6F800  00000000  00000000  | ......8......... |
  +04F0  00000000  00000000  00000000  00000000  | ................ |
  +0500  00002400  6F68D800  00000000  00000000  | ....?.Q......... |
  +0510  00000000  00000000  00000000  00000000  | ................ |
  +0520  00000000  00000000  00000000  00000000  | ................ |
  +0530  00000000  00000000  00000000  00000000  | ................ |
  +0540  00000000  00000000  00000000  00000000  | ................ |
  +0550  00000000  00000000  00000000  00000000  | ................ |
  +0560  00000000  00000000  000000              | ...........      |
                GMT-02/23/2001 15:50:35.958984  LOC-02/23/2001 11:50:35.958984
```

RACROUTE REQUEST=VERIFY

This is an example of SAF trace output for RACROUTE REQUEST=VERIFY (service type 5) PRE and POST entries. Note that the column offsets are not exact but the content has not been altered.

```
  **** GTFTRACE DISPLAY OPTIONS IN EFFECT ****
USR=ALL

**** GTF DATA COLLECTION OPTIONS IN EFFECT: ****
USRP option

               **** GTF TRACING ENVIRONMENT ****
      Release: SP7.0.3   FMID: HBB7706   System name: RACFR13
      CPU Model: 4381  Version: FF  Serial no. 111515


 USR57 F44 ASCB 00FA9B00           JOBN RACF


     Following is a formatted R_TRACE record.
      This trace record was generated by IRRTRC00 with IDENT(R_TRACE).

     Trace Identifier:            00000036
     Record Eyecatcher:           RTRACE
     Trace Type:                  RACFPRE
     Ending Sequence:             ........
     Calling address:             00000000  8B04A24E
     Requestor/Subsystem:         RSSC06 9  RACF
     Primary jobname:             RACF
     Primary asid:                0000001D
     Primary ACEEP:               00000000  006FFDC0
     Home jobname:                RACF
     Home asid:                   0000001D
     Home ACEEP:                  00000000  006FFDC0
     Task address:                00000000  006EC1A0
     Task ACEEP:                  00000000  00000000
     Time:                        B5773AAD  0E780C4B
```

```
Error class:                  ........
Service number:               00000005
RACF Return code:             00000000
RACF Reason code:             00000000
Return area address:          00000000  00000001
Parameter count:              0000000A


Area length:                  00000068

Area value:
00000000  00000000  00680200  00055800  | ................ |
0B089158  0B089160  0B08916C  00000000  | ..j...j-..j%.... |
00000000  00000068  00000000  00000000  | ................ |
00400000  00000000  00000000  00000000  | . ............... |
00000000  00000000  00000000  00000000  | ................ |
00000000  00000000  00000000  00000000  | ................ |
00000000  00000000                      | ........         |

Area length:                  0000006C

Area value:
6C0000A0  00000000  00000000  00000000  | %............... |
00000000  00000000  00000000  00000000  | ................ |
00000000  00000000  00000000  00000000  | ................ |
00000000  0B089154  00000000  00000000  | ......j......... |
00000000  00000000  00000000  00000000  | ................ |
00000000  00000000  00000000  00000000  | ................ |
00000000  00000000  00000000            | ............     |

Area length:                  00000008

Area value:
D6C6C6E2  C5E30034                      | OFFSET..         |

Area length:                  00000004

Area value:
006CF930                                | .%9.             |

Area length:                  000000A8

Area value:
C1C3C5C5  FF0000A8  02000000  00000000  | ACEE...y........ |
00000000  085CC2E8  D7C1E2E2  5C015C40  | .....*BYPASS*.*  |
40404040  40400000  0001059F  40404040  |       ......    |
40404040  00000000  00000000  00000000  |     ............ |
40404040  40404040  00000000  00000000  |         ........ |
00000000  00000000  40404040  40404040  | ........         |
00000000  00000000  00000000  00000000  | ................ |
00000000  00000000  00000000  0101059F  | ................ |
00000000  00200000  00000000  00000000  | ................ |
00000000  00000000  006CF8A0  00000000  | .........%8..... |
00000000  006CF850                      | .....%8&         |

Area length:                  00000050

Area value:
50010003  4002C000  00000000  00000000  | &... .{......... |
00000000  00000000  00000000  00000000  | ................ |
00000000  00000000  00000000  00000000  | ................ |
F0F14040  40404040  00000000  00000000  | 01      ........ |
5CC2E8D7  C1E2E25C  40404040  40404040  | *BYPASS*         |

Area length:                  00000090

Area value:
C1C3C5E7  03000000  00000000  00000000  | ACEX............ |
00000000  00000000  00000000  00000000  | ................ |
00000000  00000000  00000000  00000000  | ................ |
00000000  00000000  00000000  00000000  | ................ |
00000000  00000000  00000000  00000000  | ................ |
00000000  00000000  00000000  00000000  | ................ |
00000000  00000000  00000000  00000000  | ................ |
00000000  00000000  00000000  00000000  | ................ |
00000000  00000000  00000000  00000000  | ................ |

Area length:                  000000A8

Area value:
C1C3C5C5  FF0000A8  02000000  00000000  | ACEE...y........ |
00000000  07C9C2D4  E4E2C5D9  4004E2E8  | .....IBMUSER .SY |
```

```
   E2F14040  4040A110  4001059F  D9C1C3C6  | S1    ~. ...RACF |
   40404040  00000000  00000000  00000000  |     ............ |
   40404040  40404040  00000000  00000000  |         ........ |
   00000000  00000000  40404040  40404040  | ........         |
   00000000  006FD0E8  00000000  00000000  | .....?}Y........ |
   00000000  006FDFC8  00000000  0101059F  | .....?.H........ |
   00000000  00200000  00000000  00000000  | ................ |
   00000000  00000000  006FFB48  00000000  | .........?...... |
   00000000  006FFD70                      | .....?..         |

   Area length:                 00000050

   Area value:
   50012204  08004000  00000000  00000000  | &..... ........ |
   00000000  00000000  00000000  00000000  | ................ |
   00000000  00000000  00000000  00000000  | ................ |
   00000000  00000000  00000000  00000000  | ................ |
   C9C2D4E4  E2C5D940  E2E8E2F1  40404040  | IBMUSER SYS1     |

   Area length:                 00000090

   Area value:
   C1C3C5E7  03000000  00000000  00000000  | ACEX............ |
   00000000  00000000  00000000  00000000  | ................ |
   00000000  00000024  006FE008  00000000  | .........?\..... |
   00000000  00000000  00000000  00000000  | ................ |
   00000000  00000000  00000000  00000000  | ................ |
   00000000  00000000  00000000  00000000  | ................ |
   00000000  00000000  00000000  00000000  | ................ |
   00000000  00000000  00000000  00000000  | ................ |

   Hexadecimal dump of record follows:
+0000  00000036  D9E3D9C1  C3C54040  D9C1C3C6  | ....RTRACE  RACF |
+0010  D7D9C540  00000000  00000000  00000000  | PRE ............ |
+0020  00000000  00000000  00000000  8B04A24E  | ..............s+ |
+0030  D9E2E2C3  F0F640F9  00000000  00000000  | RSSC06 9........ |
+0040  D9C1C3C6  40404040  006EC1A0  00FA9B00  | RACF    .>A..... |
+0050  00FA9B00  0000001D  0000001D  D9C1C3C6  | ............RACF |
+0060  40404040  D9C1C3C6  40404040  006FFDC0  |     RACF    .?.{ |
+0070  006FFDC0  00000000  B5773AAD  0E780C4B  | .?.{.....:.x.K |
+0080  00000000  00000001  0000000A  00000005  | ................ |
+0090  00000068  00000000  00000000  00680200  | ...............  |
+00A0  00055800  0B089158  0B089160  0B08916C  | ......j...j-..j% |
+00B0  00000000  00000000  00000068  00000000  | ...............  |
+00C0  00000000  00400000  00000000  00000000  | ..... .......... |
+00D0  00000000  00000000  00000000  00000000  | ................ |
+00E0  00000000  00000000  00000000  00000000  | ................ |
+00F0  00000000  00000000  00000000  0000006C  | ...............% |
+0100  6C0000A0  00000000  00000000  00000000  | %............... |
+0110  00000000  00000000  00000000  00000000  | ................ |
+0120  00000000  00000000  00000000  00000000  | ................ |
+0130  00000000  0B089154  00000000  00000000  | ......j......... |
+0140  00000000  00000000  00000000  00000000  | ................ |
+0150  00000000  00000000  00000000  00000000  | ................ |
+0160  00000000  00000000  00000000  00000008  | ................ |
+0170  D6C6C6E2  C5E30034  00000004  006CF930  | OFFSET.......%9. |
+0180  000000A8  C1C3C5C5  FF0000A8  02000000  | ...yACEE...y.... |
+0190  00000000  00000000  085CC2E8  D7C1E2E2  | .........*BYPASS |
+01A0  5C015C40  40404040  40400000  0001059F  | *.*       ...... |
+01B0  40404040  40404040  00000000  00000000  |           ........ |
+01C0  00000000  40404040  40404040  00000000  | ....        .... |
+01D0  00000000  00000000  00000000  40404040  | ............     |
+01E0  40404040  00000000  00000000  00000000  |     ............ |
+01F0  00000000  00000000  00000000  00000000  | ................ |
+0200  0101059F  00000000  00200000  00000000  | ................ |
+0210  00000000  00000000  00000000  006CF8A0  | .............%8. |
+0220  00000000  00000000  006CF850  00000050  | .........%8&...& |
+0230  50010003  4002C000  00000000  00000000  | &... .{......... |
+0240  00000000  00000000  00000000  00000000  | ................ |
+0250  00000000  00000000  00000000  00000000  | ................ |
+0260  F0F14040  40404040  00000000  00000000  | 01      ........ |
+0270  5CC2E8D7  C1E2E25C  40404040  40404040  | *BYPASS*         |
+0280  00000090  C1C3C5E7  03000000  00000000  | ....ACEX........ |
+0290  00000000  00000000  00000000  00000000  | ................ |
+02A0  00000000  00000000  00000000  00000000  | ................ |
+02B0  00000000  00000000  00000000  00000000  | ................ |
+02C0  00000000  00000000  00000000  00000000  | ................ |
+02D0  00000000  00000000  00000000  00000000  | ................ |
+02E0  00000000  00000000  00000000  00000000  | ................ |
+02F0  00000000  00000000  00000000  00000000  | ................ |
+0300  00000000  00000000  00000000  00000000  | ................ |
```

```
  +0310  00000000  000000A8  C1C3C5C5  FF0000A8  | .......yACEE...y |
  +0320  02000000  00000000  00000000  07C9C2D4  | .............IBM |
  +0330  E4E2C5D9  4004E2E8  E2F14040  4040A110  | USER .SYS1    ~. |
  +0340  4001059F  D9C1C3C6  40404040  00000000  |  ...RACF    .... |
  +0350  00000000  00000000  40404040  40404040  | ........          |
  +0360  00000000  00000000  00000000  00000000  | ................. |
  +0370  40404040  40404040  00000000  006FD0E8  |         .....?}Y |
  +0380  00000000  00000000  00000000  006FDFC8  | .............?.H |
  +0390  00000000  0101059F  00000000  00200000  | ................. |
  +03A0  00000000  00000000  00000000  00000000  | ................. |
  +03B0  006FFB48  00000000  00000000  006FFD70  | .?...........?.. |
  +03C0  00000050  50012204  08004000  00000000  | ...&&..... ..... |
  +03D0  00000000  00000000  00000000  00000000  | ................. |
  +03E0  00000000  00000000  00000000  00000000  | ................. |
  +03F0  00000000  00000000  00000000  00000000  | ................. |
  +0400  00000000  C9C2D4E4  E2C5D940  E2E8E2F1  | ....IBMUSER SYS1 |
  +0410  40404040  00000090  C1C3C5E7  03000000  |     ....ACEX....  |
  +0420  00000000  00000000  00000000  00000000  | ................. |
  +0430  00000000  00000000  00000000  00000024  | ................. |
  +0440  006FE008  00000000  00000000  00000000  | .?\.............. |
  +0450  00000000  00000000  00000000  00000000  | ................. |
  +0460  00000000  00000000  00000000  00000000  | ................. |
  +0470  00000000  00000000  00000000  00000000  | ................. |
  +0480  00000000  00000000  00000000  00000000  | ................. |
  +0490  00000000  00000000  00000000  00000000  | ................. |
  +04A0  00000000  00000000                      | ........          |
               GMT-02/28/2001 19:57:22.490298  LOC-02/28/2001 15:57:22.490298

USR57 F44 ASCB 00FA9B00           JOBN RACF
    Following is a formatted R_TRACE record.
     This trace record was generated by IRRTRC00 with IDENT(R_TRACE).

    Trace Identifier:             00000036
    Record Eyecatcher:            RTRACE
    Trace Type:                   RACFPOST
    Ending Sequence:              ........
    Calling address:              00000000  8B04A24E
    Requestor/Subsystem:          RSSC06 9  RACF
    Primary jobname:              RACF
    Primary asid:                 0000001D
    Primary ACEEP:                00000000  006FFDC0
    Home jobname:                 RACF
    Home asid:                    0000001D
    Home ACEEP:                   00000000  006FFDC0
    Task address:                 00000000  006EC1A0
    Task ACEEP:                   00000000  00000000
    Time:                         B5773AAD  0E8C548B
    Error class:                  ........
    Service number:               00000005
    RACF Return code:             00000000
    RACF Reason code:             00000000
    Return area address:          00000000  0B0891C4
    Parameter count:              00000005


    Area length:                  00000068

    Area value:
    00000000  00000000  00680200  00055800  | ................ |
    0B089158  0B089160  0B08916C  00000000  | ..j...j-..j%.... |
    00000000  00000068  00000000  00000000  | ................ |
    00400000  00000000  00000000  00000000  | . ............... |
    00000000  00000000  00000000  00000000  | ................ |
    00000000  00000000  00000000  00000000  | ................ |
    00000000  00000000                      | ........         |

    Area length:                  0000006C

    Area value:
    6C0000A0  00000000  00000000  00000000  | %............... |
    00000000  00000000  00000000  00000000  | ................ |
    00000000  00000000  00000000  00000000  | ................ |
    00000000  0B089154  00000000  00000000  | ......j......... |
    00000000  00000000  00000000  00000000  | ................ |
    00000000  00000000  00000000  00000000  | ................ |
    00000000  00000000  00000000            | ............     |

    Area length:                  000000A8

    Area value:
    C1C3C5C5  FF0000A8  02000000  00000000  | ACEE...y........ |
    00000000  07C9C2D4  E4E2C5D9  4004E2E8  | .....IBMUSER .SY |
```

```
      E2F14040  4040A110  4001059F  D9C1C3C6  | S1    ~. ...RACF |
      40404040  00000000  00000000  00000000  |     ............ |
      40404040  40404040  00000000  00000000  |         ........ |
      00000000  00000000  40404040  40404040  | ........         |
      00000000  006FD0E8  00000000  00000000  | .....?}Y........ |
      00000000  006FDFC8  00000000  0101059F  | .....?.H........ |
      00000000  00200000  00000000  00000000  | ................ |
      00000000  00000000  006FFB48  00000000  | .........?...... |
      00000000  006FFD70                      | .....?..         |

      Area length:                  00000050

      Area value:
      50012204  08004000  00000000  00000000  | &..... ......... |
      00000000  00000000  00000000  00000000  | ................ |
      00000000  00000000  00000000  00000000  | ................ |
      00000000  00000000  00000000  00000000  | ................ |
      C9C2D4E4  E2C5D940  E2E8E2F1  40404040  | IBMUSER SYS1     |

      Area length:                  00000090

      Area value:
      C1C3C5E7  03000000  00000000  00000000  | ACEX............ |
      00000000  00000000  00000000  00000000  | ................ |
      00000000  00000024  006FE008  00000000  | .........?\..... |
      00000000  00000000  00000000  00000000  | ................ |
      00000000  00000000  00000000  00000000  | ................ |
      00000000  00000000  00000000  00000000  | ................ |
      00000000  00000000  00000000  00000000  | ................ |
      00000000  00000000  00000000  00000000  | ................ |

      Hexadecimal dump of record follows:
   +0000  00000036  D9E3D9C1  C3C54040  D9C1C3C6  | ....RTRACE  RACF |
   +0010  D7D6E2E3  00000000  00000000  00000000  | POST............ |
   +0020  00000000  00000000  00000000  8B04A24E  | ..............s+ |
   +0030  D9E2E2C3  F0F640F9  00000000  00000000  | RSSC06 9........ |
   +0040  D9C1C3C6  40404040  006EC1A0  00FA9B00  | RACF    .>A..... |
   +0050  00FA9B00  0000001D  0000001D  D9C1C3C6  | ............RACF |
   +0060  40404040  D9C1C3C6  40404040  006FFDC0  |     RACF    .?.{ |
   +0070  006FFDC0  00000000  B5773AAD  0E8C548B  | .?.{.....jD..... |
   +0080  00000000  0B0891C4  00000005  00000005  | ......jD........ |
   +0090  00000068  00000000  00000000  00680200  | ................ |
   +00A0  00055800  0B089158  0B089160  0B08916C  | ......j...j-..j% |
   +00B0  00000000  00000000  00000068  00000000  | ................ |
   +00C0  00000000  00400000  00000000  00000000  | ..... .......... |
   +00D0  00000000  00000000  00000000  00000000  | ................ |
   +00E0  00000000  00000000  00000000  00000000  | ................ |
   +00F0  00000000  00000000  00000000  0000006C  | ................% |
   +0100  6C0000A0  00000000  00000000  00000000  | %................ |
   +0110  00000000  00000000  00000000  00000000  | ................ |
   +0120  00000000  00000000  00000000  00000000  | ................ |
   +0130  00000000  0B089154  00000000  00000000  | ......j......... |
   +0140  00000000  00000000  00000000  00000000  | ................ |
   +0150  00000000  00000000  00000000  00000000  | ................ |
   +0160  00000000  00000000  00000000  000000A8  | ................y |
   +0170  C1C3C5C5  FF0000A8  02000000  00000000  | ACEE...y........ |
   +0180  00000000  07C9C2D4  E4E2C5D9  4004E2E8  | .....IBMUSER .SY |
   +0190  E2F14040  4040A110  4001059F  D9C1C3C6  | S1    ~. ...RACF |
   +01A0  40404040  00000000  00000000  00000000  |     ............ |
   +01B0  40404040  40404040  00000000  00000000  |         ........ |
   +01C0  00000000  00000000  40404040  40404040  | ........         |
   +01D0  00000000  006FD0E8  00000000  00000000  | .....?}Y........ |
   +01E0  00000000  006FDFC8  00000000  0101059F  | .....?.H........ |
   +01F0  00000000  00200000  00000000  00000000  | ................ |
   +0200  00000000  00000000  006FFB48  00000000  | .........?...... |
   +0210  00000000  006FFD70  00000050  50012204  | .....?.....&&... |
   +0220  08004000  00000000  00000000  00000000  | .. ............. |
   +0230  00000000  00000000  00000000  00000000  | ................ |
   +0240  00000000  00000000  00000000  00000000  | ................ |
   +0250  00000000  00000000  00000000  C9C2D4E4  | ............IBMU |
   +0260  E2C5D940  E2E8E2F1  40404040  00000090  | SER SYS1    .... |
   +0270  C1C3C5E7  03000000  00000000  00000000  | ACEX............ |
   +0280  00000000  00000000  00000000  00000000  | ................ |
   +0290  00000000  00000024  006FE008  00000000  | .........?\..... |
   +02A0  00000000  00000000  00000000  00000000  | ................ |
   +02B0  00000000  00000000  00000000  00000000  | ................ |
   +02C0  00000000  00000000  00000000  00000000  | ................ |
   +02D0  00000000  00000000  00000000  00000000  | ................ |
   +02E0  00000000  00000000  00000000  00000000  | ................ |
```

```
   +02F0  00000000  00000000  00000000  00000000  | ................ |
               GMT-02/28/2001 19:57:22.490583  LOC-02/28/2001 15:57:22.490583
```

MANAGER ALTER

This is an example of a MANAGER ALTER trace with PRE and POST entries. Note that the column offsets are not exact but the content has not been altered.

```
  **** GTFTRACE DISPLAY OPTIONS IN EFFECT ****
USR=ALL

**** GTF DATA COLLECTION OPTIONS IN EFFECT: ****
USRP option


               **** GTF TRACING ENVIRONMENT ****
      Release: SP7.0.3   FMID: HBB7706   System name: RACFR13
      CPU Model: 4381  Version: FF  Serial no. 111515


 USR57 F44 ASCB 00FACD00            JOBN IBMUSER


      is a formatted R_TRACE record.
      This trace record was generated by IRRTRC00 with IDENT(R_TRACE).

     Trace Identifier:             00000036
     Record Eyecatcher:            RTRACE
     Trace Type:                   MNGRPRE
     Ending Sequence:              ........
     Calling address:              00000000  8B000BAE
     Requestor/Subsystem:          ........  ........
     Primary jobname:              IBMUSER
     Primary asid:                 00000019
     Primary ACEEP:                00000000  006FF608
     Home jobname:                 IBMUSER
     Home asid:                    00000019
     Home ACEEP:                   00000000  006FF608
     Task address:                 00000000  006B8220
     Task ACEEP:                   00000000  00000000
     Time:                         B57740CE  B8ED7F45
     Error class:                  ........
     Service number:               00000000
     RACF Return code:             00000000
     RACF Reason code:             00000000
     Return area address:          00000000  00000000
     Parameter count:              00000006


     Area length:                  00000050

     Area value:
     08004102  0015FFDA  001468F8  00000000  | ..........8.... |
     00000000  00000000  00000000  01000000  | ................ |
     20000000  00000000  40404040  40404040  | ........         |
     00000000  00000000  00000000  00000000  | ................ |
     00000000  00000000  00000000  00000000  | ................ |

     Area length:                  00000008

     Area value:
     D6C6C6E2  C5E30004                      | OFFSET..         |

     Area length:                  00000004

     Area value:
     03D1D1D7                               | .JJP             |

     Area length:                  000000A8

     Area value:
     C1C3C5C5  FF0000A8  02000000  00000000  | ACEE...y........ |
     00000000  07C9C2D4  E4E2C5D9  4004E2E8  | .....IBMUSER .SY |
     E2F14040  4040A110  0001059F  40404040  | S1    ~.....     |
     40404040  00B28CA0  00000000  00000000  |     ............ |
     D3D6C3C1  D3C3F1F0  00000000  00000000  | LOCALC10........ |
     00000000  00000000  40404040  40404040  | ........         |
     00000000  006FDF98  00000000  00000000  | .....?.q........ |
     00000000  006FD120  00000000  0101059F  | .....?J......... |
```

```
    00000000  00200000  00000000  00000000  | ................ |
    00000000  00000000  006FF578  7F6F5000  | .........?5."?&. |
    00000000  006FDFB0                      | .....?..         |

    Area length:                  00000050

    Area value:
    50010206  0001C000  00000000  00000000  | &.....{......... |
    00000000  00000000  00000000  00000000  | ................ |
    00000000  00000000  00000000  00000000  | ................ |
    D3D6C3C1  D3C3F1F0  00000000  00000000  | LOCALC10........ |
    C9C2D4E4  E2C5D940  E2E8E2F1  40404040  | IBMUSER SYS1     |

    Area length:                  00000090

    Area value:
    C1C3C5E7  03000000  00000000  00000000  | ACEX............ |
    00000000  00000000  00000000  00000000  | ................ |
    00000000  00000024  006FF868  00000000  | .........?8..... |
    00000000  00000000  00000000  00000000  | ................ |
    00000000  00000000  00000000  00000000  | ................ |
    00000000  00000000  00000000  00000000  | ................ |
    00000000  00000000  00000000  00000000  | ................ |
    00000000  00000000  00000000  00000000  | ................ |
    00000000  00000000  00000000  00000000  | ................ |

    Hexadecimal dump of record follows:
+0000  00000036  D9E3D9C1  C3C54040  D4D5C7D9  | ....RTRACE  MNGR |
+0010  D7D9C540  00000000  00000000  00000000  | PRE ............ |
+0020  00000000  00000000  00000000  00000000  | ................ |
+0030  00000000  00000000  00000000  00000000  | ................ |
+0040  00000000  00000000  006B8220  00FACD00  | .........,b..... |
+0050  00FACD00  00000019  00000019  C9C2D4E4  | ............IBMU |
+0060  E2C5D940  C9C2D4E4  E2C5D940  006FF608  | SER IBMUSER .?6. |
+0070  006FF608  00000000  B57740CE  B8ED7F45  | .?6....... ...". |
+0080  00000000  00000000  00000006  00000000  | ................ |
+0090  00000050  08004102  0015FFDA  001468F8  | ...&...........8 |
+00A0  00000000  00000000  00000000  00000000  | ................ |
+00B0  01000000  20000000  00000000  40404040  | ............     |
+00C0  40404040  00000000  00000000  00000000  |     ............ |
+00D0  00000000  00000000  00000000  00000000  | ................ |
+00E0  00000000  00000008  D6C6C6E2  C5E30004  | ........OFFSET.. |
+00F0  00000004  03D1D1D7  000000A8  C1C3C5C5  | .....JJP...yACEE |
+0100  FF0000A8  02000000  00000000  00000000  | ...y............ |
+0110  07C9C2D4  E4E2C5D9  4004E2E8  E2F14040  | .IBMUSER .SYS1   |
+0120  4040A110  0001059F  40404040  40404040  |   ~.....        |
+0130  00B28CA0  00000000  00000000  D3D6C3C1  | ............LOCA |
+0140  D3C3F1F0  00000000  00000000  00000000  | LC10............ |
+0150  00000000  40404040  40404040  00000000  | ....        .... |
+0160  006FDF98  00000000  00000000  00000000  | .?.q............ |
+0170  006FD120  00000000  0101059F  00000000  | .?J............. |
+0180  00200000  00000000  00000000  00000000  | ................ |
+0190  00000000  006FF578  7F6F5000  00000000  | .....?5."?&..... |
+01A0  006FDFB0  00000050  50010206  0001C000  | .?.....&&.....{. |
+01B0  00000000  00000000  00000000  00000000  | ................ |
+01C0  00000000  00000000  00000000  00000000  | ................ |
+01D0  00000000  00000000  D3D6C3C1  D3C3F1F0  | ........LOCALC10 |
+01E0  00000000  00000000  C9C2D4E4  E2C5D940  | ........IBMUSER  |
+01F0  E2E8E2F1  40404040  00000090  C1C3C5E7  | SYS1    ....ACEX |
+0200  03000000  00000000  00000000  00000000  | ................ |
+0210  00000000  00000000  00000000  00000000  | ................ |
+0220  00000024  006FF868  00000000  00000000  | .....?8......... |
+0230  00000000  00000000  00000000  00000000  | ................ |
+0240  00000000  00000000  00000000  00000000  | ................ |
+0250  00000000  00000000  00000000  00000000  | ................ |
+0260  00000000  00000000  00000000  00000000  | ................ |
+0270  00000000  00000000  00000000  00000000  | ................ |
+0280  00000000  00000000  00000000            | ............     |
              GMT-02/28/2001 20:24:48.404209  LOC-02/28/2001 16:24:48.404209

USR57 F44 ASCB 00FACD00           JOBN IBMUSER


     is a formatted R_TRACE record.
     This trace record was generated by IRRTRC00 with IDENT(R_TRACE).

    Trace Identifier:             00000036
    Record Eyecatcher:            RTRACE
    Trace Type:                   MNGRPOST
    Ending Sequence:              ........
    Calling address:              00000000  8B000BAE
    Requestor/Subsystem:          ........  ........
```

```
      Primary jobname:              IBMUSER
      Primary asid:                 00000019
      Primary ACEEP:                00000000  006FF608
      Home jobname:                 IBMUSER
      Home asid:                    00000019
      Home ACEEP:                   00000000  006FF608
      Task address:                 00000000  006B8220
      Task ACEEP:                   00000000  00000000
      Time:                         B57740CE  C6AEBD85
      Error class:                  ........
      Service number:               00000000
      RACF Return code:             00000000
      RACF Reason code:             00000000
      Return area address:          00000000  00000000
      Parameter count:              00000006


      Area length:                  00000050

      Area value:
      08004102  0015FFDA  001468F8  00000000  | ...........8.... |
      00000000  00000000  00000000  01000000  | ................ |
      20000000  00000000  C2C1E2C5  40404040  | ........BASE     |
      00000000  00000000  00000000  00000000  | ................ |
      00000000  00000000  00000000  00000000  | ................ |

      Area length:                  00000008

      Area value:
      D6C6C6E2  C5E30004                      | OFFSET..         |

      Area length:                  00000004

      Area value:
      03D1D1D7                                | .JJP             |

      Area length:                  000000A8

      Area value:
      C1C3C5C5  FF0000A8  02000000  00000000  | ACEE...y........ |
      00000000  07C9C2D4  E4E2C5D9  4004E2E8  | .....IBMUSER .SY |
      E2F14040  4040A110  0001059F  40404040  | S1    ~.....     |
      40404040  00B28CA0  00000000  00000000  |     ............ |
      D3D6C3C1  D3C3F1F0  00000000  00000000  | LOCALC10........ |
      00000000  00000000  40404040  40404040  | ........         |
      00000000  006FDF98  00000000  00000000  | .....?.q........ |
      00000000  006FD120  00000000  0101059F  | .....?J......... |
      00000000  00200000  00000000  00000000  | ................ |
      00000000  00000000  006FF578  7F6F5000  | .........?5."?&. |
      00000000  006FDFB0                      | .....?..         |

      Area length:                  00000050

      Area value:
      50010206  0001C000  00000000  00000000  | &.....{......... |
      00000000  00000000  00000000  00000000  | ................ |
      00000000  00000000  00000000  00000000  | ................ |
      D3D6C3C1  D3C3F1F0  00000000  00000000  | LOCALC10........ |
      C9C2D4E4  E2C5D940  E2E8E2F1  40404040  | IBMUSER SYS1     |

      Area length:                  00000090

      Area value:
      C1C3C5E7  03000000  00000000  00000000  | ACEX............ |
      00000000  00000000  00000000  00000000  | ................ |
      00000000  00000024  006FF868  00000000  | .........?8..... |
      00000000  00000000  00000000  00000000  | ................ |
      00000000  00000000  00000000  00000000  | ................ |
      00000000  00000000  00000000  00000000  | ................ |
      00000000  00000000  00000000  00000000  | ................ |
      00000000  00000000  00000000  00000000  | ................ |
      00000000  00000000  00000000  00000000  | ................ |

      Hexadecimal dump of record follows:
    +0000  00000036  D9E3D9C1  C3C54040  D4D5C7D9  | ....RTRACE  MNGR |
    +0010  D7D6E2E3  00000000  00000000  00000000  | POST............ |
    +0020  00000000  00000000  00000000  00000000  | ................ |
    +0030  00000000  00000000  00000000  00000000  | ................ |
    +0040  00000000  00000000  006B8220  00FACD00  | .........,b..... |
    +0050  00FACD00  00000019  00000019  C9C2D4E4  | ............IBMU |
    +0060  E2C5D940  C9C2D4E4  E2C5D940  006FF608  | SER IBMUSER .?6. |
    +0070  006FF608  00000000  B57740CE  C6AEBD85  | .?6....... .F..e |
```

```
+0080  00000000  00000000  00000006  00000000  | ................ |
+0090  00000050  08004102  0015FFDA  001468F8  | ...&...........8 |
+00A0  00000000  00000000  00000000  00000000  | ................ |
+00B0  01000000  20000000  00000000  C2C1E2C5  | ............BASE |
+00C0  40404040  00000000  00000000  00000000  |     ............ |
+00D0  00000000  00000000  00000000  00000000  | ................ |
+00E0  00000000  00000008  D6C6C6E2  C5E30004  | ........OFFSET.. |
+00F0  00000004  03D1D1D7  000000A8  C1C3C5C5  | .....JJP...yACEE |
+0100  FF0000A8  02000000  00000000  00000000  | ...y............ |
+0110  07C9C2D4  E4E2C5D9  4004E2E8  E2F14040  | .IBMUSER .SYS1   |
+0120  4040A110  0001059F  40404040  40404040  |   ~.....         |
+0130  00B28CA0  00000000  00000000  D3D6C3C1  | ............LOCA |
+0140  D3C3F1F0  00000000  00000000  00000000  | LC10............ |
+0150  00000000  40404040  40404040  00000000  | ....        .... |
+0160  006FDF98  00000000  00000000  00000000  | .?.q............ |
+0170  006FD120  00000000  0101059F  00000000  | .?J............. |
+0180  00200000  00000000  00000000  00000000  | ................ |
+0190  00000000  006FF578  7F6F5000  00000000  | .....?5."?&..... |
+01A0  006FDFB0  00000050  50010206  0001C000  | .?.....&&.....{. |
+01B0  00000000  00000000  00000000  00000000  | ................ |
+01C0  00000000  00000000  00000000  00000000  | ................ |
+01D0  00000000  00000000  D3D6C3C1  D3C3F1F0  | ........LOCALC10 |
+01E0  00000000  00000000  C9C2D4E4  E2C5D940  | ........IBMUSER  |
+01F0  E2E8E2F1  40404040  00000090  C1C3C5E7  | SYS1    ....ACEX |
+0200  03000000  00000000  00000000  00000000  | ................ |
+0210  00000000  00000000  00000000  00000000  | ................ |
+0220  00000024  006FF868  00000000  00000000  | .....?8......... |
+0230  00000000  00000000  00000000  00000000  | ................ |
+0240  00000000  00000000  00000000  00000000  | ................ |
+0250  00000000  00000000  00000000  00000000  | ................ |
+0260  00000000  00000000  00000000  00000000  | ................ |
+0270  00000000  00000000  00000000  00000000  | ................ |
+0280  00000000  00000000  00000000            | ............     |
              GMT-02/28/2001 20:24:48.460550  LOC-02/28/2001 16:24:48.460550
```

Validating input

This explains how to validate input to RACF processes.

Identifying the caller of RACF

Prerequisites

- “Obtaining a system dump” on page 47
- “Formatting a system dump” on page 47

Introduction to procedure

When a RACF abend occurs, RACF might be the cause of the problem, or the program calling RACF might have passed an incorrect parameter list, which caused RACF to issue the abend. (The parameter list is the only input RACF gets from a caller.)

This describes how to determine what module issued the RACF macro.

Procedure

1. Look at the system dump and check the TCB with field TCBCMP (equal to the abend code).
2. Locate the RB that issued the RACF SVC (SVC 82, 83, 84, or 85). In this RB, get the address of the next sequential instruction from field RBOPSW. If no such RB exists, then the executing RACF service may be PC-entered. In this case, use the “Alternative procedure” on page 80.
3. Find this address in the system dump and scan backward for the eye-catcher (a module or CSECT name printed in readable form along the right margin of the dump). That is the name of the routine that issued the RACROUTE macro.

   Note: If the address of the next sequential instruction after the RACROUTE macro is not in the system dump, check an LPA (link pack area) map to determine if it is within an LPA module. If it is within an LPA module, that module is the issuer of the RACROUTE macro.

- If the name you find is not the ICHRFR00 module, then the name found caused RACF to abend. (Note that the issuer of the RACF SVC may not have caused the original error but may have merely passed bad data created by yet another module that called it. The end of the bad data chain might be several callers away.)
- If the name you find is the ICHRFR00 module, then SAF (the system authorization facility) issued the RACF macro as requested by another module that issued the RACROUTE macro.

  To determine who issued the RACROUTE macro, locate the "current task" in the system dump and check the active RBs.

  a. Find the PRB with a RACF SVC (SVC 82, 83, 84, or 85) in the RBRTPSW2 field.

  b. Find the next RB.

     – In a formatted dump that has been printed, this is the RB below the RB with the SVC.

     – On IPCS (interactive problem control system), this is the RB displayed below the RB with the SVC.
- Check the general-purpose registers in this RB. Register 1 should have the same value as the trace entry for the SVC found in Step “3.a” on page 80. Register 13 points to the save area of the ICHRFR00 module.
- For RACROUTE functions other than VERIFY or VERIFYX:

  – Find the save area of the ICHRFR00 module and make sure that in the save area, the address at offset X'4' points to the save area of the ICHSFR00 module (the previous save area).
- For RACROUTE functions VERIFY or VERIFYX:

  – Find the save area of the ICHSFR00 module and make sure that in the save area, the address at offset X'4' points to the save area of the IRRSVY00 (or IRRSVX00) module (the previous save area).

  – Find the save area of the ICHRFR00 module and make sure that in the save area, the address at offset X'4' points to the save area of the ICHSFR00 module (the previous save area).
- Find the save area of the ICHRFR00 module and make sure that in the save area, the address at offset X'4' points to the save area of the ICHSFR00 module (the previous save area).
- Find the save area of the ICHSFR00 module and make sure that in the save area, the address at offset X'4' points to the save area of the issuer of the RACROUTE macro (the previous save area).
- Find the save area of the issuer of the RACROUTE macro. In the save area of the issuer of the RACROUTE macro, offset X'C' is the contents of Register 14 of the module that issued the RACROUTE macro. Register 14 contains the address of the instruction after the RACROUTE macro.
- Find this address in the system dump and scan backward for the eye-catcher (a module or CSECT name printed in readable form along the right margin of the dump). That is the name of the routine that issued the RACROUTE macro.

Alternative procedure

1. Look at the system dump and issue the IPCS command, SUMMARY FORMAT, to format the linkage stack for the failing task or SRB. In most cases, the top entry on this linkage stack represents the current RACF service; this can be validated by taking the formatted PSW for the stack entry and scrolling backward in memory in search of a module eyecatcher that has a RACF prefix, ICH or IRR.
2. If this stack entry represents a RACF service, then general purpose register 13 should point to a standard save area, and the caller of RACF can be located by tracing backward through a chain of such areas. If these steps do not appear to have produced the caller of RACF, then you cannot determine the caller of RACF without additional support.

Checking the parameter list passed to RACF

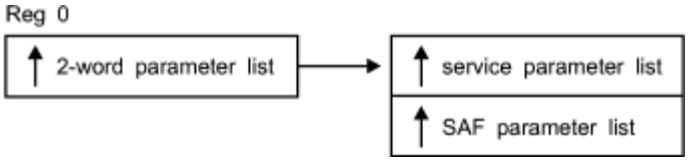
Prerequisites

- "Obtaining a system dump" on page 47
- "Formatting a system dump" on page 47.

How to check the parameter list passed to RACF

This procedure is useful only for a RACF macro request that was active when the system dump was generated.

1. Find the save area for the routine that issued the RACF macro. This can be found either in the trace-table entry for the SVC associated with the RACF macro, or in the RBs in a formatted dump. See "Identifying the caller of RACF" on page 79.
2. In the save area, find the contents of register 1.
   - If register 1 is zero, then the RACROUTE macro was issued and module ICHRFR00 issued the RACF macro. In this case, register zero points to a two-word parameter list. The first word of this parameter list points to the RACF SVC parameter list, and the second word of this parameter list points to the SAF parameter list. This is shown:

   - If register 1 is not zero, then this is the address of the parameter list that is passed with the macro request.
3. Find the address in the storage portion of the system dump. The values in the dump are the actual parameter list as seen by the system when processing the RACF request.
4. Use the description of the parameter list from "Parameter list descriptions" on page 151 to decode the actual parameter list.

   Note: For some parameter lists (such as RACROUTE REQUEST=AUTH, RACROUTE REQUEST=DEFINE, and RACROUTE REQUEST=VERIFY) the length of the parameter list varies from release to release. Check the first byte of these parameter lists for the actual length used.
5. Use these checks to determine whether the active RACF macro request had an error. You can do the checks most efficiently if you have a listing of the program that issued the macro.
   a. Did the RACF macro request that invoked RACF services use valid keyword values, including addresses?
   b. Did the caller supply a register save area for the RACF macro?
   c. Do the caller's storage areas contain correct values?

      You could check the system dump for application program storage areas that appear to have incorrect values, such as missing printable identifiers for modules or control blocks. Key storage areas are pointed to by the register save area for the RACF macro. (Incorrect values are caused by logic problems due to incorrect addresses.)

      This check is especially important for these types of unexpected abends, which might involve a logic error in either the application program that issued the RACF macro or a system routine:

      - A X'0Cx' abend, such as X'0C4'
      - A cancel (a X'x22' abend)

Alternative procedure to check the parameter list passed to RACF

After you find the caller of RACF, the first word of the parameter list pointed to from the stack entry should contain the address of the parameters passed to RACF on the request.

Searching for a matching problem

Prerequisites

1. See "Performing a preliminary search for a matching problem" on page 41 for a description of easily located problem symptoms to use to find a matching problem.
2. See "Troubleshooting the problem" on page 1 for the diagnostic procedures to follow to collect additional symptoms needed for a search.

   For problems involving system dumps, you need to follow the diagnostic procedures in "Obtaining a system dump" on page 47 and "Formatting a system dump" on page 47

How to search for a matching problem

This procedure summarizes search symptoms that you can collect by using this document's problem analysis procedures for RACF. You might want to use these symptoms to search for a matching problem in a problem database, or to report to IBM.

If you need descriptions or examples of entire search arguments, see the prerequisite procedure, "Performing a preliminary search for a matching problem" on page 41.

Scan Table 19 on page 82 for more RACF symptoms that you might want to use in your search for a matching problem (or to report to IBM). It lists some of the most probable search symptoms for problems in RACF.

Note: For abends, more symptoms such as a register/PSW (program status word) difference and a failing instruction area are automatically produced in DAEDATA output, so they are not repeated here. Notes® about the symptoms are indicated with asterisks (*) and *Note *n**, where *n* is the number of the note.

Table 19. Additional search symptoms for RACF

Description of search symptom	Free-format example	Structured example
Control block name	ACEE	FLDS/ACEE
Control block field name &rbl;&rbl;&rbl;&rbl;&rbl;*Note 1*	ACEEFLG1	FLDS/ACEEFLG1
Symptoms for later searches		
Level of failing module &rbl;&rbl;&rbl;&rbl;&rbl;*Note 2*	HRF2220 or UW09958	LVLS/HRF2220 or PTFS/UW09958

Note:

1. If you receive too many matches on your first search, consider adding a control block field name or other text from a RACF error message. (Avoid use of addresses or other values that would vary between executions of the RACF function.)
2. The ID and level at the beginning of a failing module might be pointed to by RACF's PSW at the time of the failure. For abends, the module level might be in the SDWAMLVL field and in DAEDATA output.

Application Identity Mapping Considerations

The conversion utility IRRIRA00 processes a RACF database to activate the function that controls the indexing of application identities for existing databases. IRRIRA00 is responsible for handling the non-RACF application identities that compose the alias index. The term 'alias' refers to any non-RACF application user identity (such as identities for products including Lotus Notes® for z/OS, Novell Directory Services for OS/390, and z/OS UNIX System Services) that is part of this index structure.

For more information about the IRRIRA00 utility and the stages of application identity mapping, see *z/OS Security Server RACF System Programmer's Guide*.

IRRIRA00 activates this in four stages that are controlled by the installation and lists the current stage of the active primary database. Down-level systems do not have any support for application identity mapping, so errors might occur if the RACF database is shared by systems running releases before Version 2 Release 10. Therefore, you should make your changes from your Version 2 Release 10 system only. To prevent changes being made from a down-level system inadvertently, do one of the following before running IRRIRA00:

- Migrate all your systems to Version 2 Release 10
- Use the command protection steps described in *z/OS Security Server RACF Security Administrator's Guide*

These examples point out problems that can occur when sharing the RACF database with a system running Version 2 Release 10 or higher that has the application identity mapping support at stage 1 or higher and a down-level system system:

- You ADD a user or group profile with an application identity on a down-level system. The IRRUT200 report runs successfully on the Version 2 Release 10 system, but shows that the mapping profiles were created though no alias indexes were ever created.
- You ALTER an application identity field in a user or group profile with a different application identity on the down-level system. The IRRUT200 report runs successfully on the Version 2 Release 10 system, but shows that the new mapping profiles were created though the old alias indexes that still exist. You must also issue the LISTUSER or LISTGRP command or run DBUNLOAD and compare the output to the IRRUT200 output to see differences.
- You DELETE a user or group profile with a mapping on the down-level system. The IRRUT200 report runs successfully on the Version 2 Release 10 system, but shows the mapping profiles are deleted though the old alias indexes still exist.

Additionally, if base profiles containing application identities are incorrectly manipulated from a down-level system, the alias index is not updated.

LOGREC data set entries for the alias index

In all stages, the getUMAP, getGMAP, and R-usermap callable services report unexpected ICHEINTY and RACROUTE return codes (nonzero return codes other than indicating the profile was not found) with a LOGREC entry. Additionally, if locating an alias index entry fails but a mapping profile is found for a database at Stage 2, the callable service writes an entry to the LOGREC data set with a return code of X'0000000C' and a reason code of X'00000000'. The presence of these LOGREC entries indicates that the alias index is not complete and must be corrected before moving to Stage 3. The variable data for services IRRRGM01 (GID to group name mapping service), IRRRUM01 (UID to user ID mapping service) and IRRRIM00 (application ID to RACF user ID mapping service) serves as the RACF group name or user ID and as the entry name specified on the locate request for the alias index entry.

Recovering from errors with Application Identity Mapping

Before stage 3, RACF maintains mapping profiles and functionality to ensure mapping compatibility with down-level systems that share a database with higher-level systems. This means the RACF database is susceptible to errors if a mapping profile is inadvertently deleted, or if the mapping profile is modified so that the alias index entry does not match the corresponding USER profile, as seen in the previous examples.

With application identity mapping enabled at stage 3, RACF uses an alias index rather than mapping profiles to associate users and groups with specific application identities from products such as Lotus Notes for z/OS, Novell Directory Services for OS/390, or z/OS UNIX System Services. Though unlikely, it is possible that an unexpected error could cause an association mismatch that you can identify by comparing IRRUT200 alias index output with profile information returned from LISTUSER or LISTGRP (or DBUNLOAD). This area suggests methods to correct such inconsistencies.

If the mapping profile exists, and if your database is at stage 3, no profiles in class UNIXMAP, NOTELINK, or NDSLINK should exist. If you find one, you can ignore it just as RACF does. Otherwise, you can delete it using RDELETE. For example:

```
RDELETE UNIXMAP U1
```

If the mapping profile contains lowercase letters, you cannot specify them on the RDELETE command. You must use BLKUPD or RACROUTE to delete the profile.

If your database is at stage 1 or higher and you believe the profile to be incorrect, consider one of these:

- If the alias index entry is missing, and if your database is at stage 0, you should not expect to see any alias index.
- If your database is at stage 1 or higher and you do not find any alias index entry corresponding to a specified application identity, you can regenerate the entry by altering the user or group profile with the wanted application identity. For example, you can issue:

  ```
  ALTUSER YOURID OMVS(UID(1))
  ```

- If the mapping profile associated with a user does not exist but the alias index entry does exist (at stage 1 or 2), you can remove the entry by temporarily adding the referenced profile with the indicated application identity, and then deleting the profile. For example, you can issue:

  ```
  ADDUSER YOURID OMVS(UID(1))
  DELUSER YOURID
  ```

- If an alias index entry references the incorrect user or group, but the user or group profile references a correct application identity, correct this by altering the user or group with the incorrect application identity, and then altering the user or group putting back the correct application identity. Suppose the alias index entry for the Lotus Notes for z/OS SNAME, USER1, references MYID rather than the wanted YOURID. For example, you can issue:

  ```
  ALTUSER MYID LNOTES(SNAME(USER1))
  ALTUSER YOURID LNOTES(SNAME(USER1))
  ```

Using SAF Trace to debug IBM HTTP server problems

SAF Trace may be used to debug IBM HTTP server problems related to an incorrect SAF keyring configuration. In this area, following are the steps needed to provide the appropriate output to be analyzed. Also provided is sample data for you to compare your output to.

Before you begin

You need to determine if the IBM HTTP server is using SAF keyrings.

- Verify that these lines are in httpd.main.conf:

  sslmode on

  sslport 443

  keyfile *mysslring* SAF

  If those lines do not exist in httpd.main.conf, **stop**, this diagnostic procedure does not apply.
- If the operator console says:

```
00  ICH408I USER(WEBSERV ) GROUP(IMWEB   ) NAME(####################)
       IRR.DIGTCERT.LISTRING CL(FACILITY)
       INSUFFICIENT ACCESS AUTHORITY
       ACCESS INTENT(READ   )  ACCESS ALLOWED(NONE   )
```

You must give WEBSERV READ access to IRR.DIGTCERT.LSTRING in the FACULTY class. WEBSERV is the ID assigned to the web server process.

Starting the trace

Perform these steps to use SAF Trace to debug IBM HTTP server problems related to an incorrect SAF keyring configuration.

1. Turn SAF Trace on for IRRSDL00:

   @SET TRACE(CALLABLE(TYPE(41)) JOBNAME(*))

2. Turn on GTF:

   Start gtfracf.gtf,,,noprompt

The order of trace calls

SSL calls the r_datalib (IRRSDL00) callable service. For an error free SSL connection, SAF Trace contains a pre and post record for the r_datalib functions in this order listed:

DataGetFirst
initial DataGetFirst call that fails due to bad length of parameters

DataGetFirst
the first DataGetFirst call succeeds

CheckStatus

DataGetNext
call fails due to bad length of parameters

DataGetNext
call succeeds

CheckStatus

DataGetNext
call fails because there are no more certificates in the key ring

DataAbortQuery
the call ending the query

Examples of each of the post records

Note: This trace was generated by IRRTRC00 with IDENT(R_TACE).

DataGetFirst

```
     Trace Identifier:             00000036
     Record Eyecatcher:            RTRACE
     Trace Type:                   OMVSPOST
     Ending Sequence:              ........
     Calling address:              00000000  A6168CFA
     Requestor/Subsystem:          ........  ........
     Primary jobname:              IMWEBSRV
     Primary asid:                 00000022
     Primary ACEEP:                00000000  009FFDC0
     Home jobname:                 IMWEBSRV
     Home asid:                    00000022
     Home ACEEP:                   00000000  009FFDC0
     Task address:                 00000000  009F0898
     Task ACEEP:                   00000000  00000000
     Time:                         B5F9F4C7  804A5686
     Error class:                  ........
```

```
Service number:                00000029
RACF Return code:              00000000
RACF Reason code:              00000000
Return area address:           00000000  00000000
Parameter count:               00000029
Area length:                   00000034
Area value:
2561D0C0  25615600  256155F4  25615604  | ./}{./.../.4./.. |
256155F8  25615608  256155FC  256154D1  | ./.8./.../.../.J |
2561560C  256154D2  2561D4C0  25615610  | ./.../.K./M{./.. |
A5615790                               | v/..             |
Area length:                   00000008
Area value:
D6C6C6E2  C5E30004                     | OFFSET..         |
Area length:                   00000004
Area value:
00000000                               | ....             |
Area length:                   00000008
Area value:
D6C6C6E2  C5E30008                     | OFFSET..         |
Area length:                   00000004
Area value:
00000000                               | ....             |
Area length:                   00000008
Area value:
D6C6C6E2  C5E3000C                     | OFFSET..         |
Area length:                   00000004
Area value:
00000000                               | ....             |
Area length:                   00000008
Area value:
D6C6C6E2  C5E30010                     | OFFSET..         |
```

Note: Offset 10 is the return code, which in this case is 0.

```
Area length:                   00000004
Area value:
00000000                               | ....             |
Area length:                   00000008
Area value:
D6C6C6E2  C5E30014                     | OFFSET..         |
Area length:                   00000004
Area value:
00000000                               | ....             |
Area length:                   00000008
Area value:
D6C6C6E2  C5E30018                     | OFFSET..         |
```

Note: Offset 18 is the reason code, which in this case is 0.

```
Area length:                   00000004
Area value:
00000000                               | ....             |
Area length:                   00000008
Area value:
D6C6C6E2  C5E3001C                     | OFFSET..         |
```

Note: Offset 1C is the function code, which in this case is 01.. 01 is DataGetFirst

```
Area length:                   00000001
Area value:
01                                     | .                |
Area length:                   00000008
Area value:
D6C6C6E2  C5E30020                     | OFFSET..         |
Area length:                   00000004
Area value:
00000000                               | ....             |
Area length:                   00000008
Area value:
D6C6C6E2  C5E30024                     | OFFSET..         |
Area length:                   00000001
Area value:
00                                     | .                |
Area length:                   00000008
```

```
Area value:
D6C6C6E2  C5E30028                      | OFFSET..        |
```

Note: Offset 28 is the ring name found in .conf file, which in this case is MYSSLRING.

```
Area length:                  0000000A
Area value:
09D4E8E2  E2D3D9C9  D5C7              | .MYSSLRING      |
Area length:                  00000008
Area value:
D6C6C6E2  C5E3002C                    | OFFSET..        |
Area length:                  00000004
Area value:
00000000                              | ....            |
Area length:                  00000008
Area value:
D6C6C6E2  C5E30030                    | OFFSET..        |
```

Note: Beginning with offset 30, the remaining offsets contain the function specific parameter list (FSPL).

```
Area length:                  00000048
Area value:
256157D8  00000002  00000000  000002A5  | ./.Q..........v |
256494B8  00000000  257DF558  00000000  | ..m......'5.... |
00000000  00000009  256157EC  08899999  | ........./...irr |
838599A3  81000000  0000004F  257DF870  | certa......|.'8. |
00000038  2561D5B0                    | .....N.         |
Area length:                  00000008
Area value:
D6C6C6E2  C5E30000                    | OFFSET..        |
Area length:                  00000014
Area value:
25C9A040  00000000  00000000  00000000  | .I. ........... |
00000000                              | ....            |
Area length:                  00000008
Area value:
D6C6C6E2  C5E30010                    | OFFSET..        |
```

Note: Offset 10 is the certificate.

```
Area length:                  000002A5
Area value:
308202A1  3082020A  A0030201  02020100  | .b.~.b......... |
300D0609  2A864886  F70D0101  05050030  | .....f.f7....... |
4D310B30  09060355  04061302  55533118  | (.............. |
30160603  55040A13  0F4D5920  434F4D50  | .........(...|(& |
414E592C  20494E43  31243022  06035504  | .+....+........ |
0B131B48  554D414E  20524553  4F555243  | .....(.+....|... |
45532043  45525449  46494341  5445301E  | ............... |
170D3031  30363038  30353030  30305A17  | ..............!. |
0D323030  36313030  34353935  395A304D  | ............!.( |
310B3009  06035504  06130255  53311830  | ............... |
16060355  040A130F  4D592043  4F4D5041  | ........(...|(&. |
4E592C20  494E4331  24302206  0355040B  | +....+......... |
131B4855  4D414E20  5245534F  55524345  | ....(.+....|.... |
53204345  52544946  49434154  4530819F  | ............a. |
300D0609  2A864886  F70D0101  01050003  | .....f.f7....... |
818D0030  81890281  8100B3E1  D36F32A3  | a...ai.aa...L?.t |
1ED16837  8EF47012  0F74E1F7  CA9BA924  | .J...4.....7..z. |
456081F5  6A5EDD89  35CA9674  4984F141  | .-a5.;.i..o..d1. |
E44AB5E0  F1B0DC82  75216B53  6FE7AC57  | U¢.\1..b..,.?X.. |
D73D4ECD  16161537  E5E786A1  4C0A09A9  | P.+.....VXf~<..z |
1464E572  35AD5F9C  D40F1B55  45FDF230  | ..V...¬.M.....2. |
B0228334  E6ED2A0A  6B3AD28A  B4AA789D  | ..c.W...,.K..... |
54FD84F2  1279B011  C767CCCC  26EE3ED4  | ..d2.`..G......M |
013BD21F  AEBFB074  6AE50203  010001A3  | ..K......V.....t |
81903081  8D304B06  09551D0F  0186F842  | a..a.........f8. |
010D043E  133C4765  6E657261  74656420  | ........>../.... |
62792074  68652053  65637572  65576179  | .`............/` |
20536563  75726974  79205365  72766572  | ........`....... |
20666F72  204F532F  33393020  28524143  | ..?..|.......... |
4629300E  0603551D  0F0101FF  04040302  | ................ |
0006300F  0603551D  130101FF  04053003  | ................ |
```

```
0101FF30  1D060355  1D0E0416  0414D79A  | ..............P. |
44A60457  328E9635  E655137D  52E787D4  | .w....o.W..'.XgM |
46D2300D  06092A86  4886F70D  01010505  | .K.....f.f7..... |
00038181  0052EECE  337B3893  19B9D56B  | ..aa.....#.l..N, |
AD725205  C97C4DE5  C16D13E3  1EEA02E6  | ....I@(VA_.T...W |
D00A29B1  CA6490CC  0ABAD5B9  6C61F417  | }.........N.%/4. |
95E207D5  CBE8B84B  1A7BD22A  E3BC0154  | nS.N.Y...#K.T... |
FDE66400  8A4BB3DA  0D75D127  079743F5  | .W........J..p.5 |
00ED5E58  DC4B1FF9  2937C930  E306B662  | ..;....9..I.T... |
0096DD5C  BE80A812  5F859566  15D86B0C  | .o.*..y.¬en..Q,. |
ECF12799  A58E813B  94FDF51E  45D9C3E2  | .1.rv.a.m.5..RCS |
27C64C2E  BD                          | .F<..            |
Area length:                  00000008
Area value:
D6C6C6E2  C5E30018                    | OFFSET..         |
```

Note: Offset 18 is the private_key_ptr. In this case, the length of 0 indicates that there is no key.

```
Area length:                  00000000
Area value:
Area length:                  00000008
Area value:
D6C6C6E2  C5E30028                    | OFFSET..         |
```

Note: Offset 28 is the label.

```
Area length:                  00000009
Area value:
D4E840D7  D2C940C3  C1                | MY PKI CA        |
Area length:                  00000008
Area value:
D6C6C6E2  C5E3003C                    | OFFSET..         |
Area length:                  0000004F
Area value:
304D310B  30090603  55040613  02555331  | .(.............. |
18301606  0355040A  130F4D59  20434F4D  | ..........(...|( |
50414E59  2C20494E  43312430  22060355  | &.+....+........ |
040B131B  48554D41  4E205245  534F5552  | ......(.+....|.. |
43455320  43455254  49464943  415445    | ..............  |
Area length:                  00000008
Area value:
D6C6C6E2  C5E30044                    | OFFSET..         |
Area length:                  00000038
```

Note: Offset 44 is the Subjects_DN.

```
Area value:
F0F04BD6  E47EC8E4  D4C1D54A  D9C5E2D6  | 00.OU=HUMAN¢RESO |
E4D9C3C5  E24AC3C5  D9E3C9C6  C9C3C1E3  | URCES¢CERTIFICAT |
C54BD67E  D4E84AC3  D6D4D7C1  D5E86B4A  | E.O=MY¢COMPANY,¢ |
C9D5C34B  C37EE4E2                    | INC.C=US         |
Area length:                  000000A8
Area value:
C1C3C5C5  FF0000A8  02000000  00000000  | ACEE...y........ |
00000000  07E6C5C2  E2C5D9E5  4005C9D4  | .....WEBSERV .IM |
E6C5C240  40400101  0501163F  C9D4E6C5  | WEB   ......IMWE |
C2E2D9E5  00000000  00000000  00000000  | BSRV............ |
40404040  40404040  00000000  00000000  |          ........ |
00000000  00000000  40404040  40404040  | ........         |
00000000  009FDF90  00000000  009FFA78  | ................ |
00000000  009FFAE0  00000000  0101163F  | .......\........ |
00000000  00200000  00000000  00000000  | ................ |
00000000  00000000  009FFB48  00000000  | ................ |
00000000  009FFD70                    | ........         |
Area length:                  00000050
Area value:
50012004  0000C000  00000000  00000000  | &.....{......... |
00000000  00000000  00000000  00000000  | ................ |
00000000  00000000  00000000  00000000  | ................ |
00000000  00000000  00000000  00000000  | ................ |
E6C5C2E2  C5D9E540  C9D4E6C5  C2404040  | WEBSERV IMWEB    |
Area length:                  00000090
Area value:
C1C3C5E7  03000000  00000000  00000000  | ACEX............ |
00000000  00000000  00000000  00000000  | ................ |
```

```
   00000000  00000024  009FE008  00000000  | .........\..... |
   00000000  00000000  00000000  00000000  | ................ |
   00000000  00000000  7F877040  00000000  | ........"g. .... |
   00000000  00000000  00000000  00000000  | ................ |
   00000000  00000000  00000000  00000000  | ................ |
   00000000  00000000  00000000  00000000  | ................ |
   00000000  00000000  00000000  00000000  | ................ |
```

```
   Area length:                  00000000
   Area value:
   Hexadecimal dump of record follows:
+0000  00000036  D9E3D9C1  C3C54040  D6D4E5E2  | ....RTRACE  OMVS |
+0010  D7D6E2E3  00000000  00000000  00000000  | POST............ |
+0020  00000000  00000000  00000000  00000000  | ................ |
+0030  A6168CFA  00000000  00000000  40400000  | w...........  .. |
+0040  00000000  00000000  00000000  00000000  | ................ |
+0050  00000000  00000000  009F0898  00000000  | ...........q.... |
+0060  00F55C80  00000000  00F55C80  00000022  | .5*......5*.....  |
+0070  00000022  C9D4E6C5  C2E2D9E5  C9D4E6C5  | ....IMWEBSRVIMWE |
+0080  C2E2D9E5  00000000  009FFDC0  00000000  | BSRV.......{.... |
+0090  009FFDC0  00000000  00000000  B5F9F4C7  | ...{.........94G |
+00A0  804A5686  00000000  00000000  00000000  | .¢.f............ |
+00B0  00000029  00000029  00000034  2561D0C0  | ............/}{ |
+00C0  25615600  256155F4  25615604  256155F8  | ./.../.4./.../.8 |
+00D0  25615608  256155FC  256154D1  2561560C  | ./.../.../.J./.. |
+00E0  256154D2  2561D4C0  25615610  A5615790  | ./.K./M{./..v/.. |
+00F0  00000008  D6C6C6E2  C5E30004  00000004  | ....OFFSET...... |
+0100  00000000  00000008  D6C6C6E2  C5E30008  | ........OFFSET.. |
+0110  00000004  00000000  00000008  D6C6C6E2  | ............OFFS |
+0120  C5E3000C  00000004  00000000  00000008  | ET.............. |
+0130  D6C6C6E2  C5E30010  00000004  00000000  | OFFSET.......... |
+0140  00000008  D6C6C6E2  C5E30014  00000004  | ....OFFSET...... |
+0150  00000000  00000008  D6C6C6E2  C5E30018  | ........OFFSET.. |
+0160  00000004  00000000  00000008  D6C6C6E2  | ............OFFS |
+0170  C5E3001C  00000001  01000000  08D6C6C6  | ET...........OFF |
+0180  E2C5E300  20000000  04000000  00000000  | SET............. |
+0190  08D6C6C6  E2C5E300  24000000  01000000  | .OFFSET......... |
+01A0  0008D6C6  C6E2C5E3  00280000  000A09D4  | ..OFFSET.......M |
+01B0  E8E2E2D3  D9C9D5C7  00000008  D6C6C6E2  | YSSLRING....OFFS |
+01C0  C5E3002C  00000004  00000000  00000008  | ET.............. |
+01D0  D6C6C6E2  C5E30030  00000048  256157D8  | OFFSET......./.Q |
+01E0  00000002  00000000  000002A5  256494B8  | ............v..m. |
+01F0  00000000  257DF558  00000000  00000000  | .....'5......... |
+0200  00000009  256157EC  08899999  838599A3  | ...../...irrcert |
+0210  81000000  0000004F  257DF870  00000038  | a......|.'8..... |
+0220  2561D5B0  00000008  D6C6C6E2  C5E30000  | ./N.....OFFSET.. |
+0230  00000014  25C9A040  00000000  00000000  | .....I. ........ |
+0240  00000000  00000000  00000008  D6C6C6E2  | ............OFFS |
+0250  C5E30010  000002A5  308202A1  3082020A  | ET.....v.b.~.b.. |
+0260  A0030201  02020100  300D0609  2A864886  | ............f.f |
+0270  F70D0101  05050030  4D310B30  09060355  | 7.......(....... |
+0280  04061302  55533118  30160603  55040A13  | ................ |
+0290  0F4D5920  434F4D50  414E592C  20494E43  | .(...|(&.+....+. |
+02A0  31243022  06035504  0B131B48  554D414E  | ............(.+ |
+02B0  20524553  4F555243  45532043  45525449  | ....|........... |
+02C0  46494341  5445301E  170D3031  30363038  | ................ |
+02D0  30353030  30305A17  0D323030  36313030  | ......!......... |
+02E0  34353935  395A304D  310B3009  06035504  | .....!.(........ |
+02F0  06130255  53311830  16060355  040A130F  | ................ |
+0300  4D592043  4F4D5041  4E592C20  494E4331  | (...|(&.+....+.. |
+0310  24302206  0355040B  131B4855  4D414E20  | ............(.+. |
+0320  5245534F  55524345  53204345  52544946  | ...|............ |
+0330  49434154  4530819F  300D0609  2A864886  | ......a......f.f |
+0340  F70D0101  01050003  818D0030  81890281  | 7.......a...ai.a |
+0350  8100B3E1  D36F32A3  1ED16837  8EF47012  | a...L?.t.J...4.. |
+0360  0F74E1F7  CA9BA924  456081F5  6A5EDD89  | ...7..z..-a5.;.i |
+0370  35CA9674  4984F141  E44AB5E0  F1B0DC82  | ..o..d1.U¢.\1..b |
+0380  75216B53  6FE7AC57  D73D4ECD  16161537  | ..,.?X..P.+..... |
+0390  E5E786A1  4C0A09A9  1464E572  35AD5F9C  | VXf~<..z..V...¬. |
+03A0  D40F1B55  45FDF230  B0228334  E6ED2A0A  | M.....2...c.W... |
+03B0  6B3AD28A  B4AA789D  54FD84F2  1279B011  | ,.K.......d2.`.. |
+03C0  C767CCCC  26EE3ED4  013BD21F  AEBFB074  | G......M..K..... |
+03D0  6AE50203  010001A3  81903081  8D304B06  | .V.....ta..a.... |
+03E0  09551D0F  0186F842  010D043E  133C4765  | .....f8......... |
+03F0  6E657261  74656420  62792074  68652053  | >../.....`...... |
+0400  65637572  65576179  20536563  75726974  | ....../`........ |
+0410  79205365  72766572  20666F72  204F532F  | `.........?..|.. |
+0420  33393020  28524143  4629300E  0603551D  | ................ |
+0430  0F0101FF  04040302  0006300F  0603551D  | ................ |
+0440  130101FF  04053003  0101FF30  1D060355  | ................ |
```

```
+0450  1D0E0416  0414D79A  44A60457  328E9635  | ......P..w....o. |
+0460  E655137D  52E787D4  46D2300D  06092A86  | W..'.XgM.K.....f |
+0470  4886F70D  01010505  00038181  0052EECE  | .f7.......aa.... |
+0480  337B3893  19B9D56B  AD725205  C97C4DE5  | .#.l..N,....I@(V |
+0490  C16D13E3  1EEA02E6  D00A29B1  CA6490CC  | A_.T...W}....... |
+04A0  0ABAD5B9  6C61F417  95E207D5  CBE8B84B  | ..N.%/4.nS.N.Y.. |
+04B0  1A7BD22A  E3BC0154  FDE66400  8A4BB3DA  | .#K.T....W...... |
+04C0  0D75D127  079743F5  00ED5E58  DC4B1FF9  | ..J..p.5..;....9 |
+04D0  2937C930  E306B662  0096DD5C  BE80A812  | ..I.T....o.*..y. |
+04E0  5F859566  15D86B0C  ECF12799  A58E813B  | ¬en..Q,..1.rv.a. |
+04F0  94FDF51E  45D9C3E2  27C64C2E  BD000000  | m.5..RCS.F<..... |
+0500  08D6C6C6  E2C5E300  18000000  00000000  | .OFFSET......... |
+0510  08D6C6C6  E2C5E300  28000000  09D4E840  | .OFFSET......MY  |
+0520  D7D2C940  C3C10000  0008D6C6  C6E2C5E3  | PKI CA....OFFSET |
+0530  003C0000  004F304D  310B3009  06035504  | .....|.(........ |
+0540  06130255  53311830  16060355  040A130F  | ..............  |
+0550  4D592043  4F4D5041  4E592C20  494E4331  | (...|(&.+....+.. |
+0560  24302206  0355040B  131B4855  4D414E20  | ............(.+. |
+0570  5245534F  55524345  53204345  52544946  | ...|............ |
+0580  49434154  45000000  08D6C6C6  E2C5E300  | .........OFFSET. |
+0590  44000000  38F0F04B  D6E47EC8  E4D4C1D5  | .....00.OU=HUMAN |
+05A0  4AD9C5E2  D6E4D9C3  C5E24AC3  C5D9E3C9  | ¢RESOURCES¢CERTI |
+05B0  C6C9C3C1  E3C54BD6  7ED4E84A  C3D6D4D7  | FICATE.O=MY¢COMP |
+05C0  C1D5E86B  4AC9D5C3  4BC37EE4  E2000000  | ANY,¢INC.C=US... |
+05D0  A8C1C3C5  C5FF0000  A8020000  00000000  | yACEE...y....... |
+05E0  00000000  0007E6C5  C2E2C5D9  E54005C9  | ......WEBSERV .I |
+05F0  D4E6C5C2  40404001  01050116  3FC9D4E6  | MWEB   ......IMW |
+0600  C5C2E2D9  E5000000  00000000  00000000  | EBSRV........... |
+0610  00404040  40404040  40000000  00000000  | .        ....... |
+0620  00000000  00000000  00404040  40404040  | .........       |
+0630  40000000  00009FDF  90000000  00009FFA  |  ............... |
+0640  78000000  00009FFA  E0000000  00010116  | ........\....... |
+0650  3F000000  00002000  00000000  00000000  | ................ |
+0660  00000000  00000000  00009FFB  48000000  | ................ |
+0670  00000000  00009FFD  70000000  50500120  | ............&&.. |
+0680  040000C0  00000000  00000000  00000000  | ...{............ |
+0690  00000000  00000000  00000000  00000000  | ................ |
+06A0  00000000  00000000  00000000  00000000  | ................ |
+06B0  00000000  00000000  00000000  00E6C5C2  | ............WEB |
+06C0  E2C5D9E5  40C9D4E6  C5C24040  40000000  | SERV IMWEB   ... |
+06D0  90C1C3C5  E7030000  00000000  00000000  | .ACEX........... |
+06E0  00000000  00000000  00000000  00000000  | ................ |
+06F0  00000000  00000000  24009FE0  08000000  | ...........\.... |
+0700  00000000  00000000  00000000  00000000  | ................ |
+0710  00000000  00000000  007F8770  40000000  | .........."g. ... |
+0720  00000000  00000000  00000000  00000000  | ................ |
+0730  00000000  00000000  00000000  00000000  | ................ |
+0740  00000000  00000000  00000000  00000000  | ................ |
+0750  00000000  00000000  00000000  00000000  | ................ |
+0760  00000000  00                            | .....            |
```

Note: All remaining examples contain the Function Specific Parameter List (FSPL) part of the trace record. All functions have the same parameter in the r_datalib parameter list, with different values.

CheckStatus

```
    D6C6C6E2  C5E30030                    | OFFSET..         |
    Area length:                00000008
    Area value:
    000002A5  256494B8                    | ...v..m.         |
    Area length:                00000008
    Area value:
    D6C6C6E2  C5E30004                    | OFFSET..         |
```

Note: Offset 4 is the Certificate.

```
    Area length:                000002A5
    Area value:
    308202A1  3082020A  A0030201  02020100  | .b.~.b.......... |
    300D0609  2A864886  F70D0101  05050030  | .....f.f7....... |
    4D310B30  09060355  04061302  55533118  | (............... |
    30160603  55040A13  0F4D5920  434F4D50  | .........(...|(& |
    414E592C  20494E43  31243022  06035504  | .+....+......... |
    0B131B48  554D414E  20524553  4F555243  | .....(.+....|... |
    45532043  45525449  46494341  5445301E  | ............... |
    170D3031  30363038  30353030  30305A17  | ..............!. |
    0D323030  36313030  34353935  395A304D  | ..............!.( |
    310B3009  06035504  06130255  53311830  | ............... |
```

```
    16060355  040A130F  4D592043  4F4D5041  | ........(...|(&. |
    4E592C20  494E4331  24302206  0355040B  | +....+.......... |
    131B4855  4D414E20  5245534F  55524345  | ....(.+....|.... |
    53204345  52544946  49434154  4530819F  | ..............a. |
    300D0609  2A864886  F70D0101  01050003  | .....f.f7....... |
    818D0030  81890281  8100B3E1  D36F32A3  | a...ai.aa...L?.t |
    1ED16837  8EF47012  0F74E1F7  CA9BA924  | .J...4.....7..z. |
    456081F5  6A5EDD89  35CA9674  4984F141  | .-a5.;.i..o..d1. |
    E44AB5E0  F1B0DC82  75216B53  6FE7AC57  | U¢.\1..b..,.?X.. |
    D73D4ECD  16161537  E5E786A1  4C0A09A9  | P.+.....VXf~<..z |
    1464E572  35AD5F9C  D40F1B55  45FDF230  | ..V...¬.M.....2. |
    B0228334  E6ED2A0A  6B3AD28A  B4AA789D  | ..c.W...,.K..... |
    54FD84F2  1279B011  C767CCCC  26EE3ED4  | ..d2.`..G......M |
    013BD21F  AEBFB074  6AE50203  010001A3  | ..K......V.....t |
    81903081  8D304B06  09551D0F  0186F842  | a..a.........f8. |
    010D043E  133C4765  6E657261  74656420  | ........>../.... |
    62792074  68652053  65637572  65576179  | .`............/` |
    20536563  75726974  79205365  72766572  | ........`....... |
    20666F72  204F532F  33393020  28524143  | ..?..|.......... |
    4629300E  0603551D  0F0101FF  04040302  | ................ |
    0006300F  0603551D  130101FF  04053003  | ................ |
    0101FF30  1D060355  1D0E0416  0414D79A  | ..............P. |
    44A60457  328E9635  E655137D  52E787D4  | .w....o.W..'.XgM |
    46D2300D  06092A86  4886F70D  01010505  | .K.....f.f7..... |
    00038181  0052EECE  337B3893  19B9D56B  | ..aa.....#.l..N, |
    AD725205  C97C4DE5  C16D13E3  1EEA02E6  | ....I@(VA_.T...W |
    D00A29B1  CA6490CC  0ABAD5B9  6C61F417  | }.........N.%/4. |
    95E207D5  CBE8B84B  1A7BD22A  E3BC0154  | nS.N.Y...#K.T... |
    FDE66400  8A4BB3DA  0D75D127  079743F5  | .W........J..p.5 |
    00ED5E58  DC4B1FF9  2937C930  E306B662  | ..;....9..I.T... |
    0096DD5C  BE80A812  5F859566  15D86B0C  | .o.*..y.¬en..Q,. |
    ECF12799  A58E813B  94FDF51E  45D9C3E2  | .1.rv.a.m.5..RCS |
    27C64C2E  BD                          | .F<..            |
    Area length:                  000000A8
    Area value:
    C1C3C5C5  FF0000A8  02000000  00000000  | ACEE...y........ |
    00000000  07E6C5C2  E2C5D9E5  4005C9D4  | .....WEBSERV .IM |
    E6C5C240  40400101  0501163F  C9D4E6C5  | WEB   ......IMWE |
    C2E2D9E5  00000000  00000000  00000000  | BSRV............ |
    40404040  40404040  00000000  00000000  |          ........ |
    00000000  00000000  40404040  40404040  | ........         |
    00000000  009FDF90  00000000  009FFA78  | ................ |
    00000000  009FFAE0  00000000  0101163F  | .......\........ |
    00000000  00200000  00000000  00000000  | ................ |
    00000000  00000000  009FFB48  00000000  | ................ |
    00000000  009FFD70                      | ........         |
    Area length:                  00000050
    Area value:
    50012004  0000C000  00000000  00000000  | &.....{......... |
    00000000  00000000  00000000  00000000  | ................ |
    00000000  00000000  00000000  00000000  | ................ |
    00000000  00000000  00000000  00000000  | ................ |
    E6C5C2E2  C5D9E540  C9D4E6C5  C2404040  | WEBSERV IMWEB    |
    Area length:                  00000090
    Area value:
    C1C3C5E7  03000000  00000000  00000000  | ACEX............ |
    00000000  00000000  00000000  00000000  | ................ |
    00000000  00000024  009FE008  00000000  | .........\..... |
    00000000  00000000  00000000  00000000  | ................ |
    00000000  00000000  7F877040  00000000  | ........"g. .... |
    00000000  00000000  00000000  00000000  | ................ |
    00000000  00000000  00000000  00000000  | ................ |
    00000000  00000000  00000000  00000000  | ................ |
    00000000  00000000  00000000  00000000  | ................ |
    Area length:                  00000000
    Area value:
    Hexadecimal dump of record follows:
+0000  00000036  D9E3D9C1  C3C54040  D6D4E5E2  | ....RTRACE  OMVS |
+0010  D7D6E2E3  00000000  00000000  00000000  | POST............ |
+0020  00000000  00000000  00000000  00000000  | ................ |
+0030  A6168F34  00000000  00000000  40400000  | w...........  .. |
+0040  00000000  00000000  00000000  00000000  | ................ |
+0050  00000000  00000000  009F0898  00000000  | ..........q.... |
+0060  00F55C80  00000000  00F55C80  00000022  | .5*......5*..... |
+0070  00000022  C9D4E6C5  C2E2D9E5  C9D4E6C5  | ....IMWEBSRVIMWE |
+0080  C2E2D9E5  00000000  009FFDC0  00000000  | BSRV.......{.... |
+0090  009FFDC0  00000000  00000000  B5F9F4C7  | ...{.........94G |
+00A0  80C66C46  00000000  00000000  00000000  | .F%............. |
+00B0  0000001F  00000029  00000034  2561D0C0  | ............/}{ |
+00C0  25615600  256155F4  25615604  256155F8  | ./.../.4./.../.8 |
+00D0  25615608  256155FC  256154D0  2561560C  | ./.../.../.}./.. |
+00E0  256154D2  2561D4C0  25615610  A561554C  | ./.K./M{./..v/.< |
```

```
+00F0  00000008  D6C6C6E2  C5E30004  00000004  | ....OFFSET..... |
+0100  00000000  00000008  D6C6C6E2  C5E30008  | ........OFFSET.. |
+0110  00000004  00000000  00000008  D6C6C6E2  | ............OFFS |
+0120  C5E3000C  00000004  00000000  00000008  | ET.............. |
+0130  D6C6C6E2  C5E30010  00000004  00000000  | OFFSET.......... |
+0140  00000008  D6C6C6E2  C5E30014  00000004  | ....OFFSET..... |
+0150  00000000  00000008  D6C6C6E2  C5E30018  | ........OFFSET.. |
+0160  00000004  00000000  00000008  D6C6C6E2  | ............OFFS |
+0170  C5E3001C  00000001  04000000  08D6C6C6  | ET...........OFF |
+0180  E2C5E300  20000000  04000000  00000000  | SET............. |
+0190  08D6C6C6  E2C5E300  24000000  01000000  | .OFFSET......... |
+01A0  0008D6C6  C6E2C5E3  00280000  000A09D4  | ..OFFSET.......M |
+01B0  E8E2E2D3  D9C9D5C7  00000008  D6C6C6E2  | YSSLRING....OFFS |
+01C0  C5E3002C  00000004  00000000  00000008  | ET.............. |
+01D0  D6C6C6E2  C5E30030  00000008  000002A5  | OFFSET.........v |
+01E0  256494B8  00000008  D6C6C6E2  C5E30004  | ..m.....OFFSET.. |
+01F0  000002A5  308202A1  3082020A  A0030201  | ...v.b.~.b...... |
+0200  02020100  300D0609  2A864886  F70D0101  | .........f.f7... |
+0210  05050030  4D310B30  09060355  04061302  | ....(.......... |
+0220  55533118  30160603  55040A13  0F4D5920  | ............(.. |
+0230  434F4D50  414E592C  20494E43  31243022  | .|(&.+....+..... |
+0240  06035504  0B131B48  554D414E  20524553  | .........(.+.... |
+0250  4F555243  45532043  45525449  46494341  | |............... |
+0260  5445301E  170D3031  30363038  30353030  | ................ |
+0270  30305A17  0D323030  36313030  34353935  | ..!............. |
+0280  395A304D  310B3009  06035504  06130255  | .!.(............ |
+0290  53311830  16060355  040A130F  4D592043  | ............(... |
+02A0  4F4D5041  4E592C20  494E4331  24302206  | |(&.+....+...... |
+02B0  0355040B  131B4855  4D414E20  5245534F  | ........(.+....| |
+02C0  55524345  53204345  52544946  49434154  | ................ |
+02D0  4530819F  300D0609  2A864886  F70D0101  | ..a.....f.f7... |
+02E0  01050003  818D0030  81890281  8100B3E1  | ....a...ai.aa... |
+02F0  D36F32A3  1ED16837  8EF47012  0F74E1F7  | L?.t.J...4.....7 |
+0300  CA9BA924  456081F5  6A5EDD89  35CA9674  | ..z..-a5.;.i..o. |
+0310  4984F141  E44AB5E0  F1B0DC82  75216B53  | .d1.U¢.\1..b..,. |
+0320  6FE7AC57  D73D4ECD  16161537  E5E786A1  | ?X..P.+.....VXf~ |
+0330  4C0A09A9  1464E572  35AD5F9C  D40F1B55  | <..z..V...¬.M... |
+0340  45FDF230  B0228334  E6ED2A0A  6B3AD28A  | ..2...c.W...,.K. |
+0350  B4AA789D  54FD84F2  1279B011  C767CCCC  | ......d2.`..G... |
+0360  26EE3ED4  013BD21F  AEBFB074  6AE50203  | ...M..K......V.. |
+0370  010001A3  81903081  8D304B06  09551D0F  | ...ta..a........ |
+0380  0186F842  010D043E  133C4765  6E657261  | .f8.........>../ |
+0390  74656420  62792074  68652053  65637572  | .....`.......... |
+03A0  65576179  20536563  75726974  79205365  | ../`........`... |
+03B0  72766572  20666F72  204F532F  33393020  | ......?..|...... |
+03C0  28524143  4629300E  0603551D  0F0101FF  | ................ |
+03D0  04040302  0006300F  0603551D  130101FF  | ................ |
+03E0  04053003  0101FF30  1D060355  1D0E0416  | ................ |
+03F0  0414D79A  44A60457  328E9635  E655137D  | ..P..w....o.W..' |
+0400  52E787D4  46D2300D  06092A86  4886F70D  | .XgM.K.....f.f7. |
+0410  01010505  00038181  0052EECE  337B3893  | ......aa.....#.l |
+0420  19B9D56B  AD725205  C97C4DE5  C16D13E3  | ..N,....I@(VA_.T |
+0430  1EEA02E6  D00A29B1  CA6490CC  0ABAD5B9  | ...W}.........N. |
+0440  6C61F417  95E207D5  CBE8B84B  1A7BD22A  | %/4.nS.N.Y...#K. |
+0450  E3BC0154  FDE66400  8A4BB3DA  0D75D127  | T....W........J. |
+0460  079743F5  00ED5E58  DC4B1FF9  2937C930  | .p.5..;....9..I. |
+0470  E306B662  0096DD5C  BE80A812  5F859566  | T....o.*..y.¬en. |
+0480  15D86B0C  ECF12799  A58E813B  94FDF51E  | .Q,..1.ɪv.a.m.5. |
+0490  45D9C3E2  27C64C2E  BD000000  A8C1C3C5  | .RCS.F<.....yACE |
+04A0  C5FF0000  A8020000  00000000  00000000  | E...y........... |
+04B0  0007E6C5  C2E2C5D9  E54005C9  D4E6C5C2  | ..WEBSERV .IMWEB |
+04C0  40404001  01050116  3FC9D4E6  C5C2E2D9  |    ......IMWEBSR |
+04D0  E5000000  00000000  00000000  00404040  | V.............  |
+04E0  40404040  40000000  00000000  00000000  |      ........... |
+04F0  00000000  00404040  40404040  40000000  | .....        ... |
+0500  00009FDF  90000000  00009FFA  78000000  | ................ |
+0510  00009FFA  E0000000  00010116  3F000000  | ....\........... |
+0520  00002000  00000000  00000000  00000000  | ................ |
+0530  00000000  00009FFB  48000000  00000000  | ................ |
+0540  00009FFD  70000000  50500120  040000C0  | ........&&.....{ |
+0550  00000000  00000000  00000000  00000000  | ................ |
+0560  00000000  00000000  00000000  00000000  | ................ |
+0570  00000000  00000000  00000000  00000000  | ................ |
+0580  00000000  00000000  00E6C5C2  E2C5D9E5  | .........WEBSERV |
+0590  40C9D4E6  C5C24040  40000000  90C1C3C5  |  IMWEB   ....ACE |
+05A0  E7030000  00000000  00000000  00000000  | X............... |
+05B0  00000000  00000000  00000000  00000000  | ................ |
+05C0  00000000  24009FE0  08000000  00000000  | .......\........ |
+05D0  00000000  00000000  00000000  00000000  | ................ |
+05E0  00000000  007F8770  40000000  00000000  | .....".g. ....... |
+05F0  00000000  00000000  00000000  00000000  | ................ |
+0600  00000000  00000000  00000000  00000000  | ................ |
```

```
+0610  00000000  00000000  00000000  00000000  | ................ |
+0620  00000000  00000000  00000000  00000000  | ................ |
+0630  00                                      | .                |
```

DataGetNext

The resulting example output is the same as a **DataGetFirst** example.

DataAbortQuery

Your results may contain a certificate. This particular DataAbortQuery follows a DataGetNext that failed because there were no certificates are in the key ring.

```
   D6C6C6E2  C5E30030                        | OFFSET..         |
   Area length:                  00000004
   Area value:
   256157D8                                 | ./.Q             |
   Area length:                  00000008
   Area value:
   D6C6C6E2  C5E30000                       | OFFSET..         |
   Area length:                  00000014
   Area value:
   00000000  00000000  00000000  00000000  | ................ |
   00000000                                 | ....             |
   Area length:                  000000A8
   Area value:
   C1C3C5C5  FF0000A8  02000000  00000000  | ACEE...y........ |
   00000000  07E6C5C2  E2C5D9E5  4005C9D4  | .....WEBSERV .IM |
   E6C5C240  40400101  0501163F  C9D4E6C5  | WEB   ......IMWE |
   C2E2D9E5  00000000  00000000  00000000  | BSRV............ |
   40404040  40404040  00000000  00000000  |          ........ |
   00000000  00000000  40404040  40404040  | ........         |
   00000000  009FDF90  00000000  009FFA78  | ................ |
   00000000  009FFAE0  00000000  0101163F  | .......\........ |
   00000000  00200000  00000000  00000000  | ................ |
   00000000  00000000  009FFB48  00000000  | ................ |
   00000000  009FFD70                       | ........         |
   Area length:                  00000050
   Area value:
   50012004  0000C000  00000000  00000000  | &.....{......... |
   00000000  00000000  00000000  00000000  | ................ |
   00000000  00000000  00000000  00000000  | ................ |
   00000000  00000000  00000000  00000000  | ................ |
   E6C5C2E2  C5D9E540  C9D4E6C5  C2404040  | WEBSERV IMWEB    |
   Area length:                  00000090
   Area value:
   C1C3C5E7  03000000  00000000  00000000  | ACEX............ |
   00000000  00000000  00000000  00000000  | ................ |
   00000000  00000024  009FE008  00000000  | .........\..... |
   00000000  00000000  00000000  00000000  | ................ |
   00000000  00000000  7F877040  00000000  | ........"g. .... |
   00000000  00000000  00000000  00000000  | ................ |
   00000000  00000000  00000000  00000000  | ................ |
   00000000  00000000  00000000  00000000  | ................ |
   00000000  00000000  00000000  00000000  | ................ |
   Area length:                  00000000
   Area value:
   Hexadecimal dump of record follows:
+0000  00000036  D9E3D9C1  C3C54040  D6D4E5E2  | ....RTRACE  OMVS |
+0010  D7D6E2E3  00000000  00000000  00000000  | POST............ |
+0020  00000000  00000000  00000000  00000000  | ................ |
+0030  A616A852  00000000  00000000  40400000  | w.y.........  .. |
+0040  00000000  00000000  00000000  00000000  | ................ |
+0050  00000000  00000000  009F0898  00000000  | ...........q.... |
+0060  00F55C80  00000000  00F55C80  00000022  | .5*.....5*.....  |
+0070  00000022  C9D4E6C5  C2E2D9E5  C9D4E6C5  | ....IMWEBSRVIMWE |
+0080  C2E2D9E5  00000000  009FFDC0  00000000  | BSRV.......{.... |
+0090  009FFDC0  00000000  00000000  B5F9F4C7  | ...{.........94G |
+00A0  84F95A2C  00000000  00000000  00000000  | d9!............. |
+00B0  0000001F  00000029  00000034  2561D0C0  | ............./}{ |
+00C0  25615600  256155F4  25615604  256155F8  | ./.../.4./.../.8 |
+00D0  25615608  256155FC  256154D1  2561560C  | ./.../.../.J./.. |
+00E0  256154D2  2561D4C0  25615610  A5615790  | ./.K./M{./..v/.. |
+00F0  00000008  D6C6C6E2  C5E30004  00000004  | ....OFFSET...... |
+0100  00000000  00000008  D6C6C6E2  C5E30008  | ........OFFSET.. |
+0110  00000004  00000000  00000008  D6C6C6E2  | ............OFFS |
+0120  C5E3000C  00000004  00000000  00000008  | ET.............. |
+0130  D6C6C6E2  C5E30010  00000004  00000000  | OFFSET.......... |
+0140  00000008  D6C6C6E2  C5E30014  00000004  | ....OFFSET...... |
+0150  00000000  00000008  D6C6C6E2  C5E30018  | ........OFFSET.. |
+0160  00000004  00000000  00000008  D6C6C6E2  | ............OFFS |
```

```
+0170  C5E3001C  00000001  03000000  08D6C6C6  | ET..........OFF |
+0180  E2C5E300  20000000  04000000  00000000  | SET............. |
+0190  08D6C6C6  E2C5E300  24000000  01000000  | .OFFSET......... |
+01A0  0008D6C6  C6E2C5E3  00280000  000A09D4  | ..OFFSET.......M |
+01B0  E8E2E2D3  D9C9D5C7  00000008  D6C6C6E2  | YSSLRING....OFFS |
+01C0  C5E3002C  00000004  00000000  00000008  | ET.............. |
+01D0  D6C6C6E2  C5E30030  00000004  256157D8  | OFFSET......./.Q |
+01E0  00000008  D6C6C6E2  C5E30000  00000014  | ....OFFSET...... |
+01F0  00000000  00000000  00000000  00000000  | ................ |
+0200  00000000  000000A8  C1C3C5C5  FF0000A8  | .......yACEE...y |
+0210  02000000  00000000  00000000  07E6C5C2  | ............WEB |
+0220  E2C5D9E5  4005C9D4  E6C5C240  40400101  | SERV .IMWEB   .. |
+0230  0501163F  C9D4E6C5  C2E2D9E5  00000000  | ....IMWEBSRV.... |
+0240  00000000  00000000  40404040  40404040  | ........        |
+0250  00000000  00000000  00000000  00000000  | ................ |
+0260  40404040  40404040  00000000  009FDF90  |         ........ |
+0270  00000000  009FFA78  00000000  009FFAE0  | ...............\ |
+0280  00000000  0101163F  00000000  00200000  | ................ |
+0290  00000000  00000000  00000000  00000000  | ................ |
+02A0  009FFB48  00000000  00000000  009FFD70  | ................ |
+02B0  00000050  50012004  0000C000  00000000  | ...&&.....{..... |
+02C0  00000000  00000000  00000000  00000000  | ................ |
+02D0  00000000  00000000  00000000  00000000  | ................ |
+02E0  00000000  00000000  00000000  00000000  | ................ |
+02F0  00000000  E6C5C2E2  C5D9E540  C9D4E6C5  | ....WEBSERV IMWE |
+0300  C2404040  00000090  C1C3C5E7  03000000  | B   ....ACEX.... |
+0310  00000000  00000000  00000000  00000000  | ................ |
+0320  00000000  00000000  00000000  00000024  | ................ |
+0330  009FE008  00000000  00000000  00000000  | ..\............. |
+0340  00000000  00000000  00000000  00000000  | ................ |
+0350  7F877040  00000000  00000000  00000000  | "g. ............ |
+0360  00000000  00000000  00000000  00000000  | ................ |
+0370  00000000  00000000  00000000  00000000  | ................ |
+0380  00000000  00000000  00000000  00000000  | ................ |
+0390  00000000  00000000  00000000            | ............     |
```

Using the trace to find problems

If SSL does not start, or starts but SSL handshake does not occur, turn on trace, and restart the web server. Calls to r_datalib occur when the web server is being started. You should be able to tell from this list what went wrong based on the trace you received. For these trace examples, important trace information is followed by an explanatory note.

- **First DataGetFirst contains a return code of 8 with a reason code of 54**

  All of the trace results offset 8 are not shown. The sample trace follows:

```
Area value:
D6C6C6E2  C5E30010                    | OFFSET..         |
```

  Note: Offset 10 is the return code

```
Area length:                 00000004
Area value:
00000008                              | ....             |
```

  Note: Hex 8 in the return code value

```
Area length:                 00000008
Area value:
D6C6C6E2  C5E30014                    | OFFSET..         |
Area length:                 00000004
Area value:
00000000                              | ....             |
Area length:                 00000008
Area value:
D6C6C6E2  C5E30018                    | OFFSET..         |
```

  Note: Offset 18 is the reason code

```
D6C6C6E2  C5E30018                    | OFFSET..         |
Area length:                 00000004
```

```
     Area value:
     00000054                                | ....            |
```

Note: Hex 54 is the reason code value

54 in hex is equal to a reason code of 80 which for this function is reason: Profile for ring_name not found. The next step for fixing this problem would be to look at the trace at offset 28 and compare the ring name found there with the one in the keyfile area of the .conf file. An example of output at offset 28 is the first DataGetFirst previous example. Check that the ring name is correct. Note, ring names are case-sensitive and must be owned by the web server daemon user ID (WEBSRV by default). This can be checked using the RACDCERT LISTRING TSO command.

- **SSL doesn't start on the server and there are no traces.**

  Check the keyfile like in the http.main.conf file. It should look like this:

  ```
  keyfile MYSSLRING SAF
  ```

  If the SAF positional parameter is missing, then System SSL treats the name as a GSKKYMAN key file, not a RACF keyring, hence there is no trace. If the keyfile really is a RACF keyring, add the SAF positional parameter.

- **Second DataGetNext has a reason code 2C.**

  Hex 2C is reason code 44: Record not found. This means that there is a problem with one of the two certificates that are attached to the ring. To find out what exactly is wrong, some more research is needed.

  In order for SSL to function properly the key ring needs:

  – to be owned by the ID who assigned to the web server process
  – to contain a signing CERTAUTH certificate with CERTAUTH usage
  – to contain a PERSONAL certificate owned by the ID who is assigned to the web server process. This certificate must also be signed by the CERTAUTH certificate and be marked DEFAULT.
  – to have neither certificate marked NOTRUST

  To learn what could be wrong a RACDCERT should be issued:

  ```
  RACDCERT ID(webserver-user-id) LISTRING(MYSSLRING)
  ```

  WEBSERV is the name of the ID who assigned to the web server process. MYSSLRING is the name that is listed in the keyfile area of the .conf file. The output should be similar to this example:

  ```
  Ring:
        >MYSSLRING<
    Certificate Label Name             Cert Owner     USAGE      DEFAULT
    -------------------------------    ------------   --------   -------
    MY PKI CA                          CERTAUTH       CERTAUTH     NO
    SSL CERT                           ID(WEBSERV)    PERSONAL     YES
  ```

  Looking at this output verifies most of these conditions.

  The output: "IRRD114I Ring MYSSLRING does not exist." indicates that the ring in the .conf file does not exist, or that it is not owned by the ID assigned to the web server process.

  To check the signer of the PERSONAL certificate is the CERTAUTH certificate and that neither certificate is marked NOTRUST, other commands must be used.

  Using the previous certificate label names, the commands and output are listed:

  Command:

  ```
  RACDCERT ID(WEBSERV) LIST(LABEL('SSL CERT'))
  ```

Output:

```
Label: SSL CERT
Certificate ID: 2QfmxcLixdnl4uLTQMPF2eNA
Status: TRUST
Start Date: 2001/06/25 00:00:00
End Date:   2002/06/25 23:59:59
Serial Number:
     >01<
Issuer's Name:
     >OU=HUMAN RESOURCES CERTIFICATE.O=MY COMPANY, INC.C=US<
Subject's Name:
     >CN=WWW.MYCOMPANY.COM.O=MY COMPANY INC.L=MILLBROOK.SP=NEW YORK.C=US<
Signing Algorithm: sha1RSA
Key Type: RSA
Key Size: 1024
Private Key: YES
Ring Associations:
  Ring Owner: WEBSERV
  Ring:
     >MYSSLRING<
```

Command:

```
RACDCERT CERTAUTH LIST(LABEL('MY PKI CA'))
```

Output:

```
Label: MY PKI CA
Certificate ID: 2QiJmZmDhZmjgdToQNfSyUDDwUBA
Status: TRUST
Start Date: 2001/06/08 00:00:00
End Date:   2020/06/09 23:59:59
Serial Number:
     >00<
Issuer's Name:
     >OU=HUMAN RESOURCES CERTIFICATE.O=MY COMPANY, INC.C=US<
Subject's Name:
     >OU=HUMAN RESOURCES CERTIFICATE.O=MY COMPANY, INC.C=US<
Key Usage: CERTSIGN
Signing Algorithm: sha256RSA
Key Type: RSA
Key Size: 2048
Private Key: YES
Ring Associations:
  Ring Owner: WEBSERV
  Ring:
     >MYSSLRING<
```

Note: Both certificates have TRUST status and the PERSONAL certificate was issued by the CERTAUTH certificate.

Compare your results with the examples provided to evaluate your situation for common IBM HTTP Server related problems that are related to an incorrect SAF keyring configuration. If you need more assistance when you are done, you might want to consider calling the IBM support center.

Chapter 3. Reporting RACF problems to IBM

Prerequisites

See “Troubleshooting the problem” on page 1 and the diagnostic procedures for this type of problem.

You also need to understand the procedures for reporting a problem to the IBM support center.

Introduction to reporting RACF problems to IBM

This describes the information that you need to collect before you report any RACF problems to IBM. The procedure lists only the information that the IBM support center is most likely to request.

When you report a problem, you need to describe your system and the problem that you experienced. The people at the IBM support center use this information to see whether your problem is already known to IBM and to check whether a fix is available.

Note: If you have the Information/MVS or Information/Access product, or a similar problem-search tool, you can do checking for a duplicate known problem yourself. “Performing a preliminary search for a matching problem” on page 41 and “Searching for a matching problem” on page 82 have advice on which RACF symptoms to use in search arguments for search tools.

If you report a new, unique problem, you need to provide additional problem information.

How to report RACF problems to IBM

Before calling IBM, collect the information called for in the worksheet for your type of problem. See Table 20 on page 97 through Table 28 on page 103 for the applicable worksheet.

Consider collecting all the information about the worksheets before reporting the problem, to speed up your phone conversation if the support center needs the information.

The support center people usually want you to provide the information in the same order as in the checklist.

Recommendations:

1. Before you call, you need to have a listing or an interactive display ready that shows any problem information that you obtained from a system dump or another data set.
2. For the most efficient phone conversations with IBM, consider having IPCS (Interactive Problem Control System) ready for interactive dump displays during the conversation. “Formatting a system dump” on page 47 has advice about using IPCS.

For the complete list of diagnostic procedures for each type of problem, see “Troubleshooting tables for problems in RACF” on page 7.

Table 20. Information to gather before calling IBM with abend problems

Information to gather before calling IBM with abend problems

Table 20. Information to gather before calling IBM with abend problems (continued)

Information to gather before calling IBM with abend problems
You can use this worksheet to write on directly or as a guide for making your own notes elsewhere. 1. Level of z/OS: _____ (Example: z/OS Version 1 Release 1) 2. RACF is enabled for sysplex communication: _____ • non-data sharing mode _____ • data sharing mode _____ • read-only mode _____ 3. System is configured as an RRSF node: _____ • local mode _____ • remote mode _____ 4. PTF level of abending module: _____ (Example: UW27135) 5. Installation exits installed (which exits are used and what they do): 6. User modifications, if any (which modules are modified and what the modifications do): 7. **Any** recent changes (such as maintenance applied, Security Server options changed, new program products, new hardware): 8. A formatted system dump with options: `SUMMARY SUMDUMP FORMAT CURRENT TRACE LOGDATA LPAMAP CPUDATA SYMPTOMS` 9. The message that contains the abend code: _____ **Complete** message text: 10. Assembler listing, if source is available. 11. If a hardware problem is suspected, EREP (error recording and editing program) records.

Table 21. Information to gather before calling IBM with documentation problems

Information to gather before calling IBM with documentation problems
You can use this worksheet to write on directly or as a guide for making your own notes elsewhere. 1. Level of z/OS: _____ (Example: z/OS Version 1 Release 1) 2. RACF is enabled for sysplex communication: _____ • non-data sharing mode _____ • data sharing mode _____ • read-only mode _____ 3. System is configured as an RRSF node: _____ • local mode _____ • remote mode _____ 4. Installation exits installed (which exits are used and what they do): 5. User modifications, if any (which modules are modified and what the modifications do): 6. Publication title and order number (with dash level): 7. Page number or topic number: _____

Table 22. Information to gather before calling IBM with output problems

Information to gather before calling IBM with output problems
You can use this worksheet to write on directly or as a guide for making your own notes elsewhere. 1. Level of z/OS: _____ (Example: z/OS Version 1 Release 1) 2. RACF is enabled for sysplex communication: _____ • non-data sharing mode _____ • data sharing mode _____ • read-only mode _____ 3. System is configured as an RRSF node: _____ • local mode _____ • remote mode _____ 4. Installation exits installed (which exits are used and what they do): 5. User modifications, if any (which modules are modified and what the modifications do): 6. **Any** recent changes (such as maintenance applied, RACF options changed, new program products, new hardware): 7. Description of RACF request that caused incorrect output (who did it—user's attributes, routine running, or both—and what was done): 8. IBM reference material that explains what correct output should look like: 9. Online or printed copy of output, or other indications that the output is incomplete: 10. If available, please attach example of good output (before problem occurred). 11. If a report writer problem, please attach a listing of the SMF (system management facilities) records.

Table 23. Information to gather before calling IBM with logon problems

Information to gather before calling IBM with logon problems

Table 23. Information to gather before calling IBM with logon problems (continued)

Information to gather before calling IBM with logon problems
You can use this worksheet to write on directly or as a guide for making your own notes elsewhere. **Note:** Before filling out this worksheet, make sure the users are not "revoked". 1. Level of z/OS: _____ (Example: z/OS Version 1 Release 1) 2. RACF is enabled for sysplex communication: _____ • non-data sharing mode _____ • data sharing mode _____ • read-only mode _____ 3. System is configured as an RRSF node: _____ • local mode _____ • remote mode _____ 4. Installation exits installed (which exits are used and what they do): 5. User modifications, if any (which modules are modified and what the modifications do): 6. **Any** recent changes (such as maintenance applied, RACF options changed, new program products, new hardware): 7. How many users cannot log on? _____ 8. Are there any aspects that the users having logon problems share, such as group memberships or authorities? 9. When users attempt to log on, what message, if any, is displayed? 10. If applicable, have SMF records related to the logon problem available.

Table 24. Information to gather before calling IBM with loop problems

Information to gather before calling IBM with loop problems

Table 24. Information to gather before calling IBM with loop problems (continued)

Information to gather before calling IBM with loop problems
You can use this worksheet to write on directly or as a guide for making your own notes elsewhere. 1. Level of z/OS: _____ (Example: z/OS Version 1 Release 1) 2. RACF is enabled for sysplex communication: _____ • non-data sharing mode _____ • data sharing mode _____ • read-only mode _____ 3. System is configured as an RRSF node: _____ • local mode _____ • remote mode _____ 4. Installation exits installed (which exits are used and what they do): 5. User modifications, if any (which modules are modified and what the modifications do): 6. **Any** recent changes (such as maintenance applied, RACF options changed, new program products, new hardware): 7. What external symptoms indicate that there is a loop (such as multiple copies of output or poor performance)? 8. Description of routines involved in loop, with addresses of instructions, if available: 9. What conditions the loop occurs in: 10. Whether loop is ___ enabled or ___ disabled. 11. If loop is enabled, please include a standalone system dump. 12. If applicable, please have available online or (preferably) printed output.

Table 25. Information to gather before calling IBM with message problems

Information to gather before calling IBM with message problems
You can use this worksheet to write on directly or as a guide for making your own notes elsewhere. 1. Level of z/OS: _____ (Example: z/OS Version 1 Release 1) 2. RACF is enabled for sysplex communication: _____ • non-data sharing mode _____ • data sharing mode _____ • read-only mode _____ 3. System is configured as an RRSF node: _____ • local mode _____ • remote mode _____ 4. Installation exits installed (which exits are used and what they do): 5. User modifications, if any (which modules are modified and what the modifications do): 6. **Any** recent changes (such as maintenance applied, RACF options changed, new program products, new hardware): 7. **Complete** message text, including any related messages that immediately precede or follow it: 8. Name of routine running when message is issued:

Table 26. Information to gather before calling IBM with performance problems

Information to gather before calling IBM with performance problems
You can use this worksheet to write on directly or as a guide for making your own notes elsewhere. 1. Level of z/OS: _____ (Example: z/OS Version 1 Release 1) 2. RACF is enabled for sysplex communication: _____ • non-data sharing mode _____ • data sharing mode _____ • read-only mode _____ 3. System is configured as an RRSF node: _____ • local mode _____ • remote mode _____ 4. Installation exits installed (which exits are used and what they do): 5. User modifications, if any (which modules are modified and what the modifications do): 6. **Any** recent changes (such as maintenance applied, RACF options changed, new program products, new hardware): 7. Description of the conditions in which degradation occurs: 8. Any console messages (please attach).

Table 27. Information to gather before calling IBM with return code problems

Information to gather before calling IBM with return code problems

Table 27. Information to gather before calling IBM with return code problems (continued)

Information to gather before calling IBM with return code problems

You can use this worksheet to write on directly or as a guide for making your own notes elsewhere.

1. Level of z/OS: _____ (Example: z/OS Version 1 Release 1)
2. RACF is enabled for sysplex communication: _____
   - non-data sharing mode _____
   - data sharing mode _____
   - read-only mode _____
3. System is configured as an RRSF node: _____
   - local mode _____
   - remote mode _____
4. Installation exits installed (which exits are used and what they do):
5. User modifications, if any (which modules are modified and what the modifications do):
6. **Any** recent changes (such as maintenance applied, RACF options changed, new program products, new hardware):
7. The return code you expected: _____

   The return code you received: _____
8. The RACF macro request that caused the return code (including the parameters passed):
9. The order number (with dash level) and page number or topic number of your copy of the document in which you found the description of the return code:

   Order number: _____ Page number or topic number: _____
10. Assembler listing of program.

Table 28. Information to gather before calling IBM with wait problems

Information to gather before calling IBM with wait problems

You can use this worksheet to write on directly or as a guide for making your own notes elsewhere.

1. Level of z/OS: _____ (Example: z/OS Version 1 Release 1)
2. RACF is enabled for sysplex communication: _____
   - non-data sharing mode _____
   - data sharing mode _____
   - read-only mode _____
3. System is configured as an RRSF node: _____
   - local mode _____
   - remote mode _____
4. Installation exits installed (which exits are used and what they do):
5. User modifications, if any (which modules are modified and what the modifications do):
6. **Any** recent changes (such as maintenance applied, RACF options changed, new program products, new hardware):
7. PSW (program status word): _____
8. Source and assembler listing of routine in which wait occurred, if available.

Procedure

Chapter 4. Troubleshooting your RACF database

This topic describes the organization of the RACF database and explains how you can troubleshoot and repair it by using some of the RACF utilities.

With z/OS V2R5, a RACF data set can be a VSAM linear data set, if the conditions identified in The RACF database section in the *z/OS Security Server RACF Security Administrator's Guide* are met. The internal structure of the data set of a RACF VSAM data set is identical to a non-VSAM RACF data set. That is, the internal structure of the data set (ICB, templates, BAM blocks, index blocks, data blocks, free space) is the same. The diagnostic techniques described in this chapter do not change if you are using a VSAM RACF data set.

If you encounter a problem with a RACF VSAM database, diagnostic information is provided in message IRR05417I. Additionally, look for accompanying messages, such as those which begin with the message prefix "IEC" and use the information in those messages to determine the reason for the failure. If no accompanying messages appear, contact IBM Support.

You can get a better understanding of the contents of your RACF database by running the database unload utility, IRRDBU00. If you unload your RACF database to a sequential file (using the IRRDBU00 utility), you might uncover problems in your database that you find only when reading the profile. This utility reads every profile.

For more information about using the IRRDBU00 utility, see *z/OS Security Server RACF Macros and Interfaces* and *z/OS Security Server RACF Security Administrator's Guide*.

You might also find it advantageous to use the IRRUT200 RACF database verification utility program to assist in locating problems in your RACF database. For more details, see *z/OS Security Server RACF System Programmer's Guide*.

Format of the RACF database

This describes the format of the RACF database.

Each RACF database is a single extent data set that is made up of 4 KB blocks and must be cataloged.

A RACF database consists of several types of records:

Header block (inventory control block, ICB)
Is the first block in a RACF database and provides a general description of the database.

Templates
Contain mappings of the entity records for the RACF database.

Segment table block
Contains mappings of individual segments from within a template.

BAM (block availability mask) blocks
Determine allocation of space within the RACF database.

Index blocks
Locate entity records (profiles) in the RACF database.

If you are using application identity mapping and it is in stage 1, 2, or 3, there is an alternative alias index to consider. This alias index correlates an application identity (a set field that is defined within a non-base segment within the RACF templates, such as the OMVS UID field) to a base profile (the user or group profile which has an application identity field set to a particular value).

Profiles (entity records)
Contain descriptions of the attributes and authorities for every entity that is defined to RACF. These entities are:

- User profiles

- Group profiles
- DASD data set profiles
- Profiles for resources that are defined by entries in the class descriptor table

Figure 4 on page 106 illustrates the format of the database.

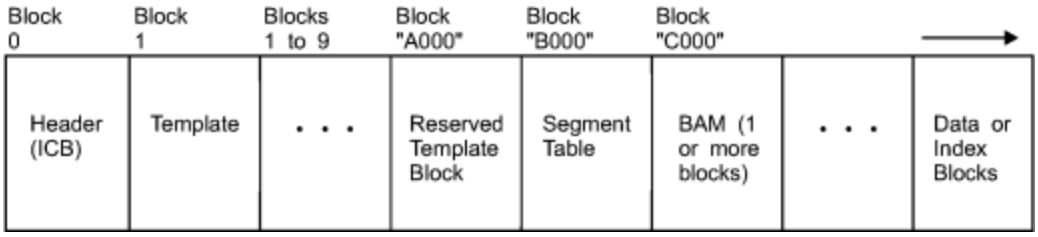


Figure 4. Format of the RACF database

Relative byte addresses (RBA) up to the first BAM block are identified by their position. The need for more BAM blocks is driven by the size of the database. See “BAM block header” on page 107 for more information.

Any RBAs that are after the BAMs consist of unassigned blocks, index blocks, and data blocks. A profile's segment data may begin at any 256-byte slot with a 4 K data block and then continue in consecutive slots, through consecutive RBAs. Therefore, it is possible that a field definition (which is what constitutes the contents of a profiles segment data) starts in one RBA, and completes upon another, possibly spanning many RBAs. The field definition might then deposit a byte of any value in the first byte of the first slot of an RBA.

If initially it might seem a false positive for an index block (X'8A'), a forceful approach read the RACF database might not be able to determine the RBA type by just checking the first byte of the RBA. The index structure and the sequence set provide the information to find a profiles segment data, in its assigned 256-byte slot with a data block. The logical length of the profile indicates the number of consecutive 256-byte slots that are needed.

Therefore, values you might use to locate blocks are:

- X'02' - segment table
- X'00' - BAM
- X'83' - data
- X'8A' - index
- X'C3®' - empty block

Header block (ICB)

The ICB (inventory control block) has a relative byte address of zero. RACF uses the ICB to locate the other blocks in a RACF database. Each RACF database has an ICB, but RACF uses only the ICB for the master primary data set when determining the setting of options.

See Appendix A, “The Inventory Control Block (ICB),” on page 243 for the fields shown in the data area, ICB.

Templates

IRRTEMP2 contains database profile mappings. When your installation creates the database, IRRMIN00 reads the mappings, formats them, and writes them to the RACF database.

RACF provides a template for each type of profile (user, group, data set, and general resource). Each profile consists of a BASE segment, plus other potential segments, for example, the DFP segment. In turn, each template contains a number that corresponds to the type of profile that it is mapping. The template maps the fields that are contained in each segment of the profile by describing the field name and field length.

If a template block is continued into another block, the last 3 bytes of the template block contains the block number of the template expansion block. This figure illustrates how the templates would look if Template 2 expanded into Block *xxx* of the database:

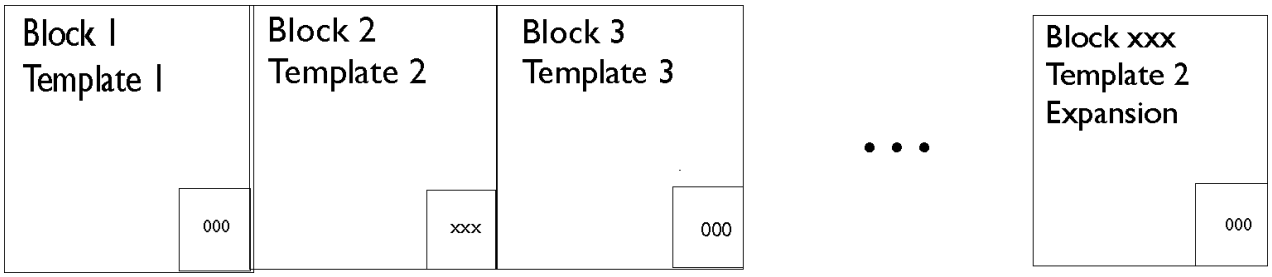


Figure 5. Template blocks

Attention:

Do not modify the template definitions that IBM supplies, or unpredictable results might occur because they are not supported.

Segment table block

The segment table block contains mappings of individual segments from within a template. These describe the segments associated with the profiles.

BAM blocks

A block availability mask (BAM) block is a 4KB block that describes the availability of the corresponding blocks in a RACF database. Each BAM block starts with a header followed by a table of two-byte "block masks".

The BAM high-water mark contained in the ICB is the address of the BAM block from which segments were last allocated or deallocated. This BAM block is used first when starting a search for space for a new allocation.

BAM block header

The BAM block header contains this information:

Bytes
Description

Bytes 0-5:
Relative byte address (RBA) of the previous BAM block (or 0 if this is the first block).

Bytes 6-11:
RBA of the next BAM block (or 0 if this is the last block).

Bytes 12-17:
RBA of the first 4KB block whose space this BAM block defines.

Bytes 18-19:
Number of 4KB blocks whose space this BAM block defines. This is also the number of two-byte block masks which follow. A BAM block can correspond to a maximum of 2038 blocks of 4096 bytes each.

BAM block masks

Following the BAM block header is a table of block masks. A block mask is a two-byte field (16 bits) that corresponds to one 4KB block in the RACF database. The first block mask corresponds to the block whose RBA is defined in Bytes 12-17 of the BAM header. Subsequent block masks in the BAM correspond successively to the 4KB blocks following this RBA. Bytes 18-19 of the BAM header contains the number of block masks in this BAM.

Each bit in the block mask corresponds, successively, to a 256-byte slot within the mask's corresponding 4KB block. If a bit is set to 1, its corresponding slot in the block is free. Otherwise, this slot is in use. Figure 6 on page 108 is an example of a block mask for a block whose first six slots are in use.

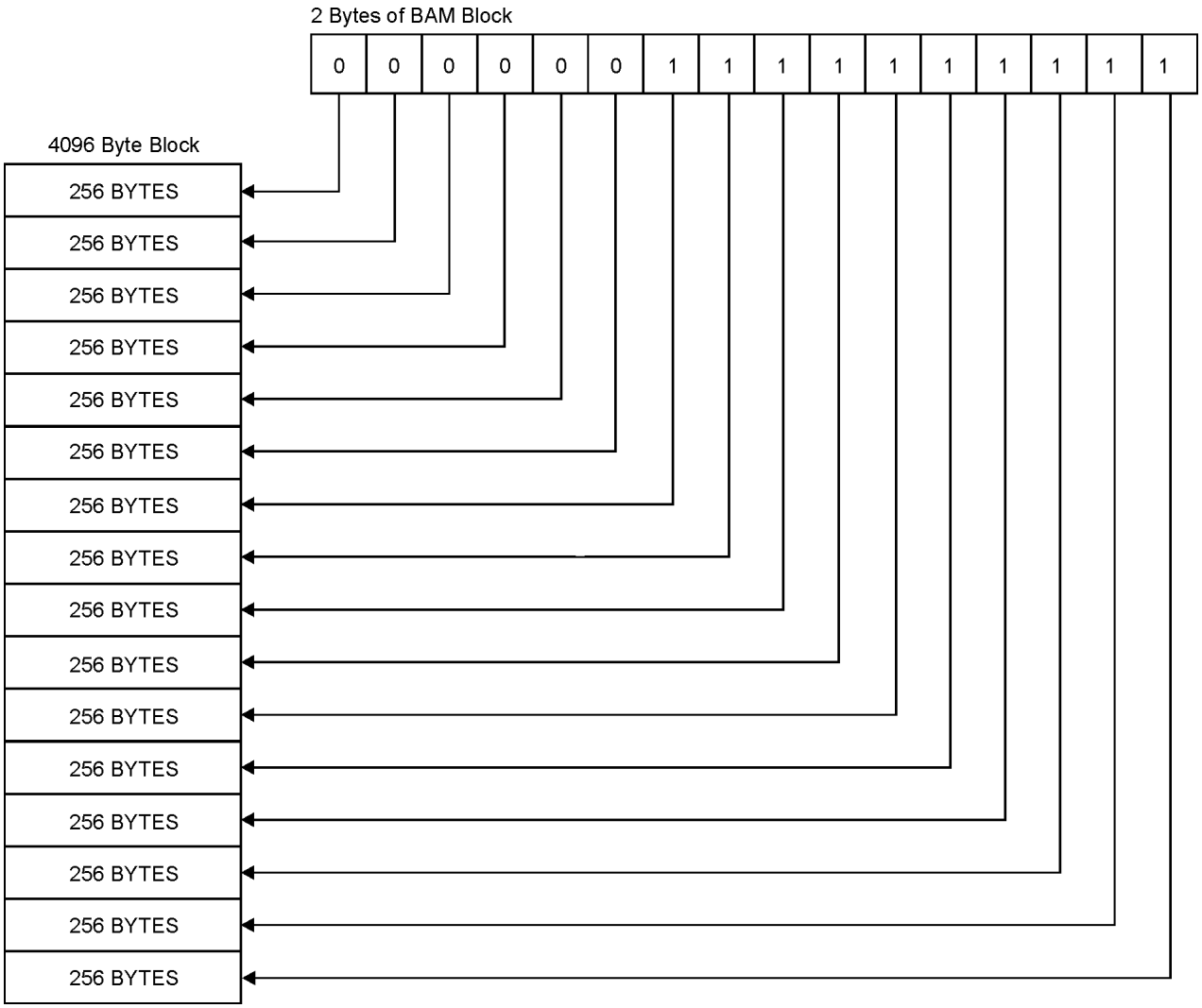


Figure 6. Block masks of the RACF database

Index blocks

RACF uses a multilevel index set to locate profiles. There can be up to 10 levels on a RACF database. All index searches begin with the highest level index block, whose RBA (relative byte address) is contained in the ICB (inventory control block). At every level but the lowest, the first entry in a block that is equal to or alphabetically greater than the requested profile name is used to reach the next lower-level index block. If no entry is greater than or equal to the profile name, the index search continues with the RBA pointed to by the last index entry in the block being searched. This situation can occur because of previous index-entry deletions.

The lowest level of index blocks (level 1) is known as the sequence set. Index entries in the sequence set contain the RBAs of the actual profile segments. Each block then points to the next block in succession.

Note: The pointer in the ICB that is pointing to the first block in the sequence set is for use by BLKUPD if the database index structure is corrupted. The sequence set pointer is also used by IRRUT400 when reorganizing the database.

Figure 7 on page 109 is an example of part of a RACF index, and it illustrates the path RACF uses to find the profile named FAL.N. This example assumes 3 levels. The level-1 blocks for the first and third level-2 blocks are partially shown in the figure.

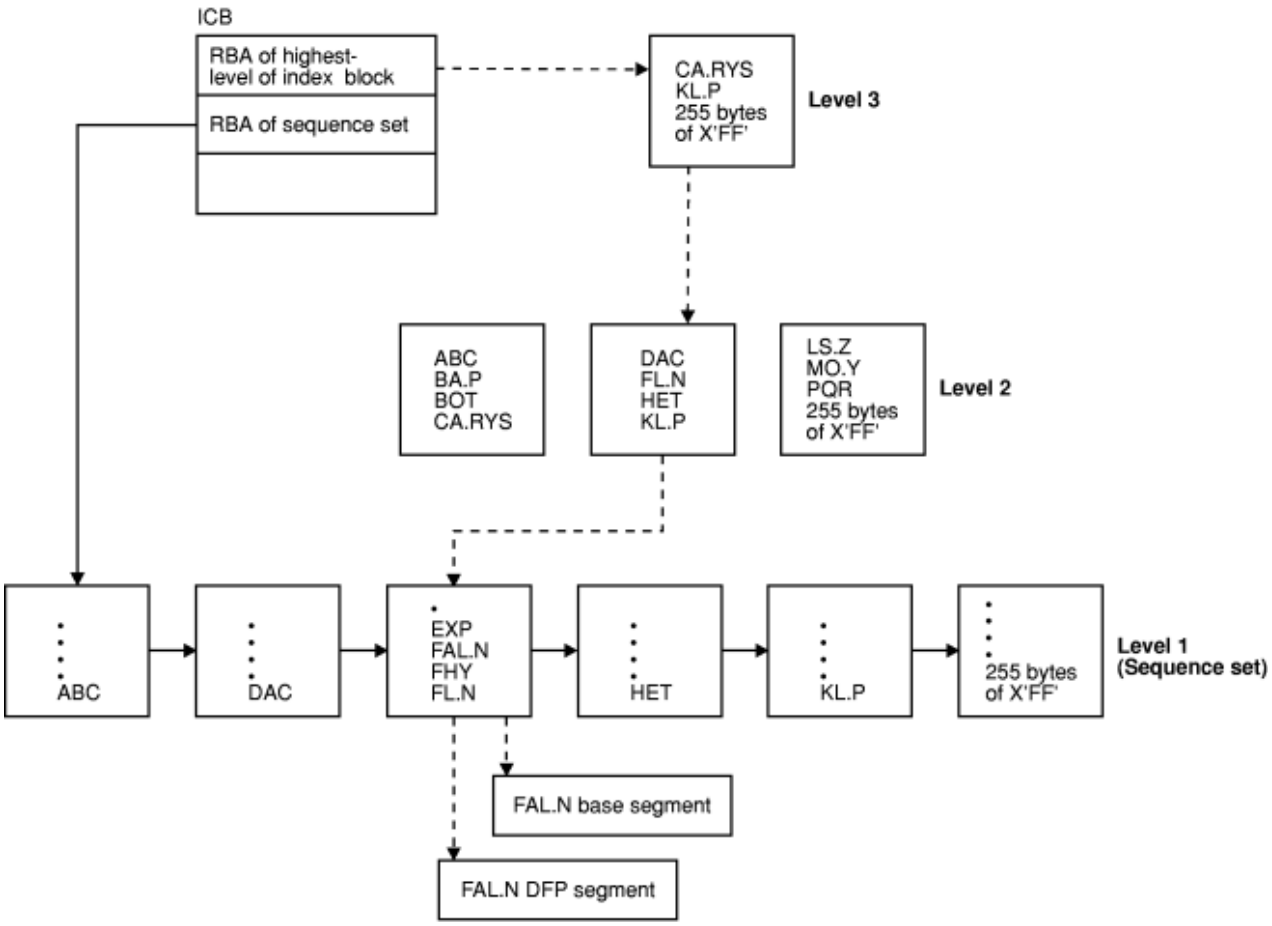


Figure 7. Example of a RACF Index

Alias index blocks

A companion to the existing index structure is the alias index structure. For alias indexes, there are four notable differences to consider:

- There are alternate anchors in the ICB
- There is a new format identifier
- There is a new index entry identifier
- The index entries in the sequence set of the alias index contain a base profile data area that names the base profile that corresponds to the particular alias index.

Overview

An index block (regular or alias) is a 4 KB block whose overview is composed of these parts:

- Header
- Table of index entries.
- Sequence set chain pointer entry. For level-1 index blocks only. Follows last index entry.
- X'0C' Delimiter
- (At end of block) Table of 2-byte offsets to the index entries.

Figure 8 on page 110 is an example of an overview of a level-1 index block. The first 14 bytes is the header. In this example, *n* (in the header) is the number of index entries. Offset X'0EF8' is the sequence set chain pointer entry (which follows the last index entry). This offset value is also found in the header.

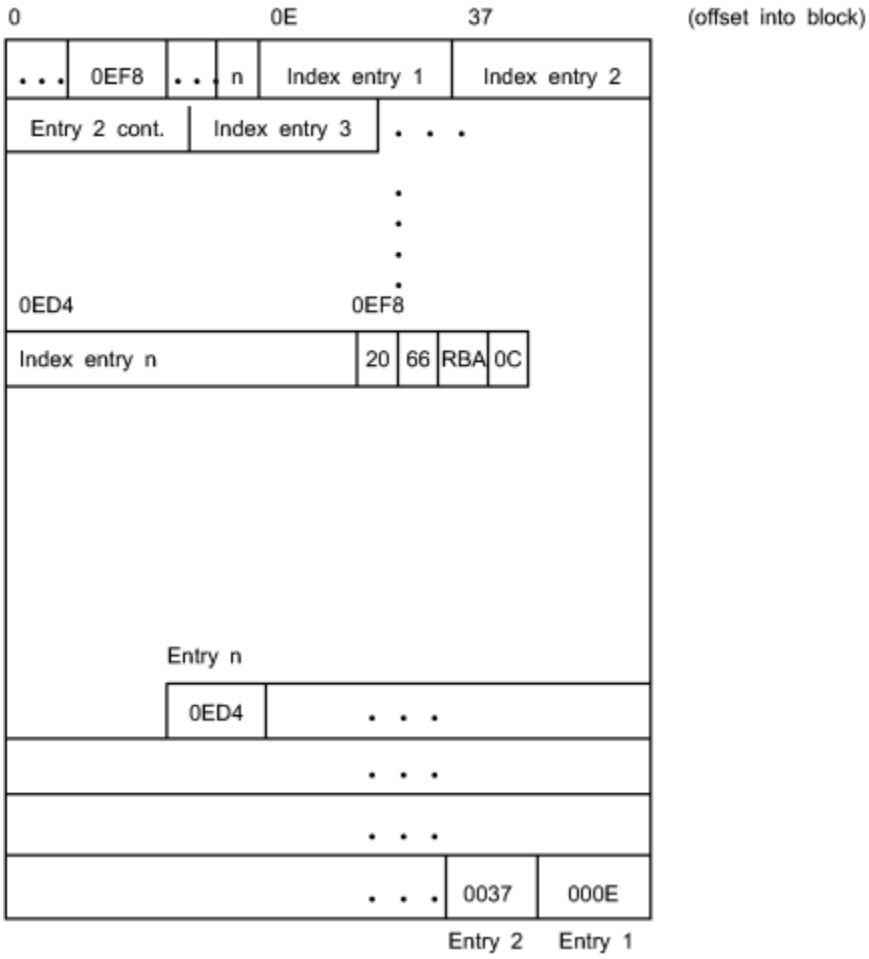


Figure 8. Example overview of a level-1 index block

Index block header

The format of an index block header in the index block is as follows:

Bytes
Description

Byte 0:
X'8A' Index block identifier

Bytes 1-2:
X'1000' Length of the index block

Byte 3:
X'4E' Index block identifier

Byte 4:
X'00' Format identifier for regular index block; X'01' Format identifier for alias index block

Byte 5:
Index-level number (01-0A)

Bytes 6-7:
Offset to the last entry in the index block. In a level-1 index block, the last entry is a sequence set chain pointer entry. In an upper-level index block, the last entry is the last index entry of the block.

Bytes 8-9:
Offset to free space in the index block

Bytes 10-11:
Offset to a table of index-entry offsets

Bytes 12-13:
Count of index entries in the block. This is also equivalent to the count of entries in the table of index-entry offsets (the bottom part of the block).

Figure 9 on page 111 shows an example of the index block header set up for the example of the level-1 index block in Figure 8 on page 110.

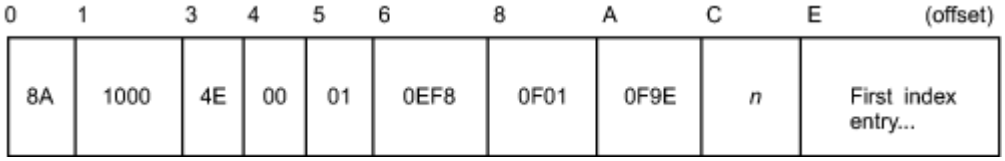


Figure 9. Example of an index block header

Index entries

Following the index block header is a table of index entries. The format of an index entry is:

Bytes
Description

Byte 0:
Entry identifier, as follows:

X'21'
Normal index entry (non-duplicate entry). In a set of duplicate index entries, this identifier marks the last index entry of the set.

X'22'
Duplicate index entry. This entry is a duplicate of the next index entry. Index entries can be duplicates of one another until one marked by the X'21' identifier is reached.

A duplicate index entry can exist only in a level-1 index block and only for data set profiles with the same name.

X'23'
Alias index entry.

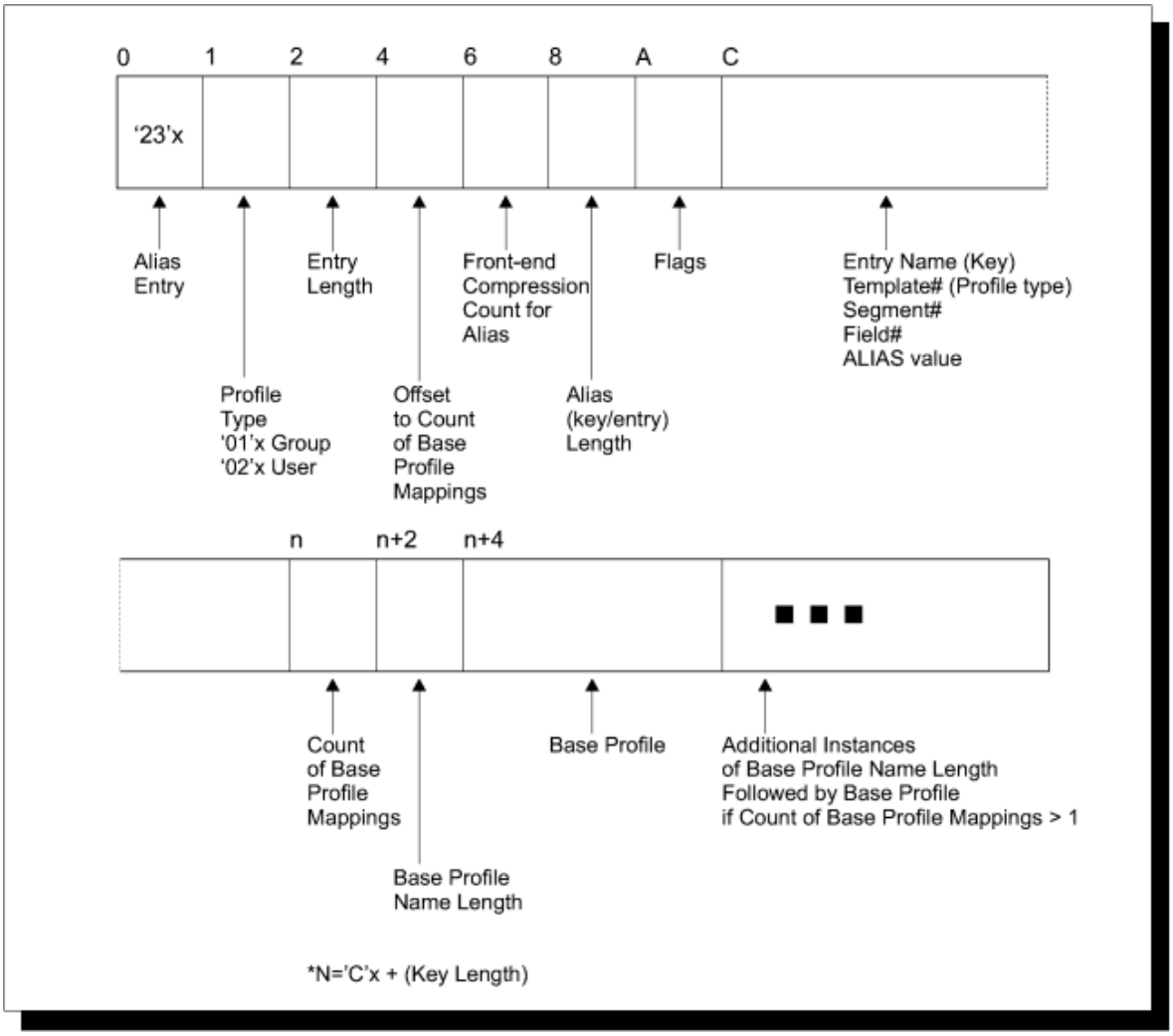


Figure 10. Example overview of an alias index block

Byte 1:
Type of profile for this index entry, as follows:

X'01'
Group

X'02'
User

X'04'
Data set

X'05'
General resource

Bytes 2-3:
Length of this index entry

Bytes 4-5:
Offset from the beginning of the entry to the segment data area for regular index entries, and offset from the beginning of the entry to the base profile data area for alias index.

Bytes 6-7:
Front-end compression count. Number of leftmost characters that are suppressed from the profile name in the index entry's name field. To save space in the database, RACF suppresses the leftmost characters that are identical to those in the preceding index entry.

To find the suppressed characters, trace the index entries backward to the one with front-end compression count of zero. In this index entry, all leftmost characters would still be intact in its name field.

The first index entry of the index block always has a front-end compression count of zero. A name that is the same as one in the previous entry might be totally compressed.

Figure 11 on page 113 shows an example of front-end compression counts.

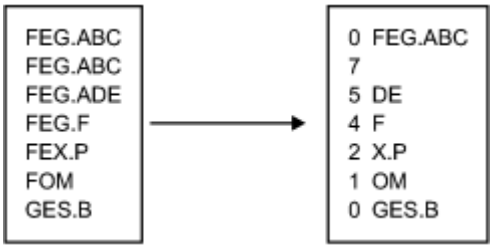


Figure 11. Example of front-end compression in an index block

Note: Not all entry names are necessarily compressed at all times. For example, when a new name is added, it might not be compressed until the database is reformatted (such as through the IRRUT400 utility).

Bytes 8-9:
Length of index entry name

Bytes 10-11:
Reserved

Bytes 12-*n*:
Index entry name, 0-255 bytes. Bytes 8-9 contain the actual length.

An index entry name corresponds to either a profile name or application identity, but it does not always match letter by letter or byte by byte. It might be compressed (see Bytes 6-7, for example), or some of its characters might be substituted by RACF for internal purposes (such as to save space or to facilitate the search).

For an alias index entry, the first 3 bytes of the entry name are non-EBCDIC and specifies the particular alias field within the template as follows:

- **Byte 0**: Template identifier. Describes the type of base profile that is associated with this alias entry.
- **Byte 1**: Segment identifier. See Table 29 on page 114.
- **Byte 2**: Field identifier. See the template descriptions in *z/OS Security Server RACF Macros and Interfaces* for field identifiers.

In an upper-level (that is, not level 1) index block, the index entry name might correspond to the entry name of the last index entry of the next-lower level block. Its rightmost characters might be compressed if RACF determines that they are not essential to finding the correct index block at the next-lower block.

An index entry name of 255 bytes of X'FF' signifies the end of index blocks for that level number.

Bytes *n*+1:
Segment data area. In an upper-level regular or alias index block, the segment data area has this format:

Byte 0:
X'62' Identifier byte

Bytes 1-6:
RBA of the next-lower-level index block

In a regular level-1 index block, the segment data area has this format:

Byte 0:
Number of segment-data fields in this data area. This is also the number of segments in this profile.

Bytes 1-7:
Segment-data field. This field is repeated successively for each segment in the profile. The format of a segment-data field is:

Byte 0:
Segment Identifier. See Table 29 on page 114 for a list of identifiers. Profile type is indicated in Byte 1 of the index entry.

Bytes 1-6:
RBA of the segment.

In an alias level-1 index block, the segment data area has this format:

Bytes 0–1:
Number of base profile fields in this data area.

Bytes 2–*n*:
Base profile data field. This field is repeated successively for each base profile reference for this application identity.

Bytes 2–3
Denotes the 1–8 character length of the user or group profile name immediately following this field.

Bytes 4–*m*:
Base profile name.

Table 29. Segment identifiers

Group	User	Data set	General
• X'01'—BASE • X'02'—DFP • X'03'—OMVS • X'04'—OVM • X'05'—TME • X'06'—CSDATA	• X'01'—BASE • X'02'—DFP • X'03'—TSO • X'04'—CICS • X'05'—LANGUAGE • X'06'—OPERPARM • X'07'—WORKATTR • X'08'—OMVS • X'09'—NETVIEW • X'0A'—DCE • X'0B'—OVM • X'0C'—LNOTES • X'0D'—NDS • X'0E'—KERB • X'0F'—PROXY • X'10'—EIM • X'11'—CSDATA	• X'01'—BASE • X'02'—DFP • X'03'—TME • X'03'—CSDATA	• X'01'—BASE • X'02'—SESSION • X'03'—DLFDATA • X'04'—SSIGNON • X'05'—STDATA • X'06'—SVFMR • X'07'—CERTDATA • X'08'—TME • X'09'—KERB • X'0A'—PROXY • X'0B'—EIM • X'0C'—ALIAS • X'0D'—CDTINFO • X'0E'—ICTX • X'0F'—CFDEF • X'10'—SIGVER • X'11'—ICSF • X'12'—MFA • X'13'—MFPOLICY • X'14'—CSDATA • X'15'—IDTPARMS • X'16'—JES

Figure 12 on page 115 shows an example of a normal index entry in a level-1 index block for a user profile named USR26, which contains three segments: Base, TSO, and DFP.

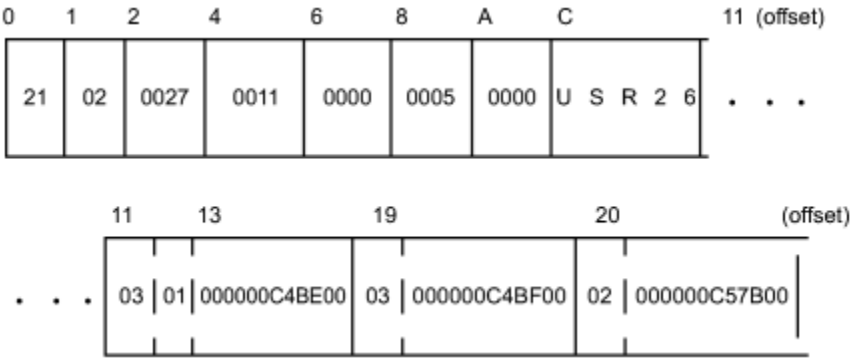


Figure 12. Example of index entry for USR26 with a base, TSO, and DFP segment

Sequence set chain pointer entry

A sequence set chain pointer entry is an entry that references to the next level-1 index block. This entry follows the last index entry of the index block and is applicable only in level-1 index blocks. It has this format:

Bytes
Description

Bytes 0-1:
X'2066' Sequence Set Chain Pointer Entry Identifier

Bytes 2-7:
RBA of next level-1 index block (0 if last one of the set)

An example of a sequence set chain pointer entry is illustrated in Figure 8 on page 110.

X'0C' delimiter

A 1-byte X'0C' delimiter marks the end of the list of entries in an index block. In an upper-level index block, it follows the last index entry. In a level-1 index block, it follows the sequence set chain pointer entry.

The X'0C' delimiter is then followed by a series of zeros until the beginning of the table of index-entry offsets at the end of the block.

Index-Entry offset table

At the end of an index block is a table of two-byte offsets to the index entries. This table starts from the bottom of the block and grows inward so that the last offset of the table is the offset for index entry 1, the next-to-last offset of the table is for index entry 2, and so on. RACF uses this table to perform a search on index entries.

In the example in Figure 8 on page 110, note that the last offset of the table is the offset of the first index entry and the first one of the table is the offset of the last index entry.

Profiles

The profiles, or entity records, contain the actual descriptions of the attributes and authorities for every entity (users, groups, DASD data sets, and resource classes defined in the class descriptor table) defined to RACF. The number in the entry-type field identifies the type of profile and corresponds to the number of the template that maps this type of profile.

The record data (including the type of profile that it is) follows the header. This data consists of the fields that are mapped by a template. See the template description corresponding to each type of profile for the contents of these fields. The template descriptions are in *z/OS Security Server RACF Macros and Interfaces*.

Because there can be duplicate entry names in different classes, the RACF manager adds a class identifier to the beginnings of general-resource entry names (for example; DASDVOL -, TAPEVOL -, or TERMINAL- for DASD volumes, tape volumes, or terminals, respectively). General-resource class names that are not eight characters in length are padded with trailing blanks.

Note: If you define a profile and use generic characters such as (*) to add members to the profile, RLIST RESGROUP does not return any of the matching profiles in its output because it does not support generic matches.

Generic profile names have the first period in a DATASET profile replaced by X'01', and the dash in the class identifier for general-resource classes replaced by X'02'. Although these expanded names are transparent to the user, they appear when using the block update utility command or the IRRUT200 utility. You also need to be aware of them when constructing a database range table.

When a tape volume profile is initially created, RACF places the tape volume serial in the volume list of the profile. RACF creates an index entry and profile name in the standard way. If another tape volume is to be added (creating a volume set), RACF adds its volume serial to the volume list in the profile and creates an index entry for the volume that points to the profile.

For example, if there are six tape volumes in a tape volume set, there are six index entries pointing to the same profile and six volume serials in the profile's volume list. When a tape volume is deleted, RACF removes the volume serial from the volume list in the profile and deletes the index entry. The profile name does not change, even if the volume after which the profile might have been named is deleted.

It is possible to have a profile name of TAPEVOL -TAPE01 without having a corresponding index entry and without having TAPE01 in the volume serial list.

The database profiles consist of segments that are made up of fields and repeat groups that follow a record header. The record header, the field structure, and the repeat group structure are described in this document.

- **Record header**

  The record header consists of these fields:

  Bytes
  Description

  Byte 0:
  X'83' Record identifier.

  Bytes 1-4:
  Physical length of this record, in bytes.

  Bytes 5-8:
  Logical record length. Length, in bytes, of the portion of the record that actually contains data.

  Bytes 9-16:
  Segment name.

  Bytes 17-18:
  Length of profile name (in Byte 20).

  Byte 19:
  Reserved

  Bytes 20+n:
  Profile name. Its length is specified in Bytes 17-18.

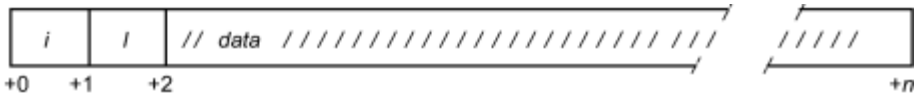
- **Field structure**

  Following the record header is a set of one or more segment data fields. Although the potential total length of these fields is 2GB, that might be limited by the type of DASD and the amount of virtual storage available to the user.

  All data fields are in a variable-length format. The first byte is the field ID. See the template descriptions in *z/OS Security Server RACF Macros and Interfaces* for field identifiers. Where indicated in the template descriptions, the field can be a member of a repeat group.

If the field is not a member of a repeat group, the field ID is followed by a field indicating the length of the data that follows. If the high-order bit of the length field is zero, this field is one byte and the data can be up to 127 bytes; if the high-order is set, the length field is 4 bytes (for example, X'80000022'), and the data can be up to 2 to the 31st power.

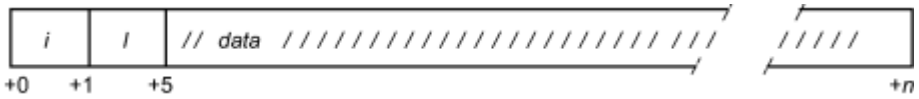
This figure shows the structure of a profile field when the length field is one byte. The structure is:



where:

- *i* is the field ID. *i* is 1 byte.
- *l* is the length of the data that follows. *l* is 1 byte.
- The high-order bit of this field is zero.
- *data* is 1 to 127 bytes of data.

This figure shows the structure of a profile field when the length field is 4 bytes. The structure is:



where:

- *i* is the field ID. *i* is 1 byte.
- *l* is the length of the data that follows. *l* is 4 bytes.
- The high-order bit of this field is set to one.
- *data* is up to 2^{31} bytes of data.

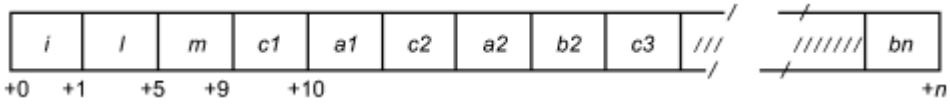
- **Repeat group structure**

  A repeat group structure contains a set of fields that are part of a repeat group. It contains all occurrences of a repeat group and, for each occurrence, it contains the count of fields and every field in the occurrence, *including those with null values.*

  The field ID (first byte) is associated with the entire repeat group. Fields in a repeat group structure do not have individual ID fields.

  Each repeat group is limited to 64KB of installation data.

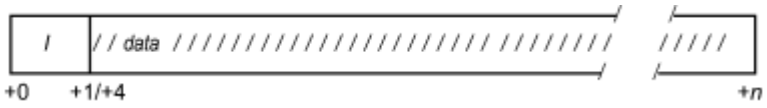
  The structure of a repeat group is:



  where:

  - *i* is the field ID. *i* is 1 byte.
  - *l* is the length of the entire repeat group. *l* is 4 bytes.
  - *m* is the count of repeat group occurrences. *m* is 4 bytes.
  - *c1, c2,* and *c3* are counts of fields in each occurrence. Each is 1 byte.
  - *a1, a2,* and *b2* are data, up to 2^{31} bytes each.

  Each data field has this format:



where:

- *l* is the length of data that follows. If its high-order bit is zero, *l* is one byte and the data length can be up to 127 bytes. If its high-order bit is set to one, *l* is 4 bytes.
- *data* is up to 2^{31} bytes of data.

When displaying a profile with the BLKUPD command, you might want to have the RACF database templates available for reference. These can be found in *z/OS Security Server RACF Macros and Interfaces*. They include a list of the field IDs in numeric order. The field IDs are in decimal in *z/OS Security Server RACF Macros and Interfaces* but in hexadecimal when displayed by the BLKUPD command.

Example of a profile

Figure 13 on page 119 shows a profile and identifies the record header, the first profile field structure, and the first repeat group structure. The field IDs of the other fields are shaded. You can use BLKUPD to display a profile. See “Block update command for a RACF database (BLKUPD)” on page 120.

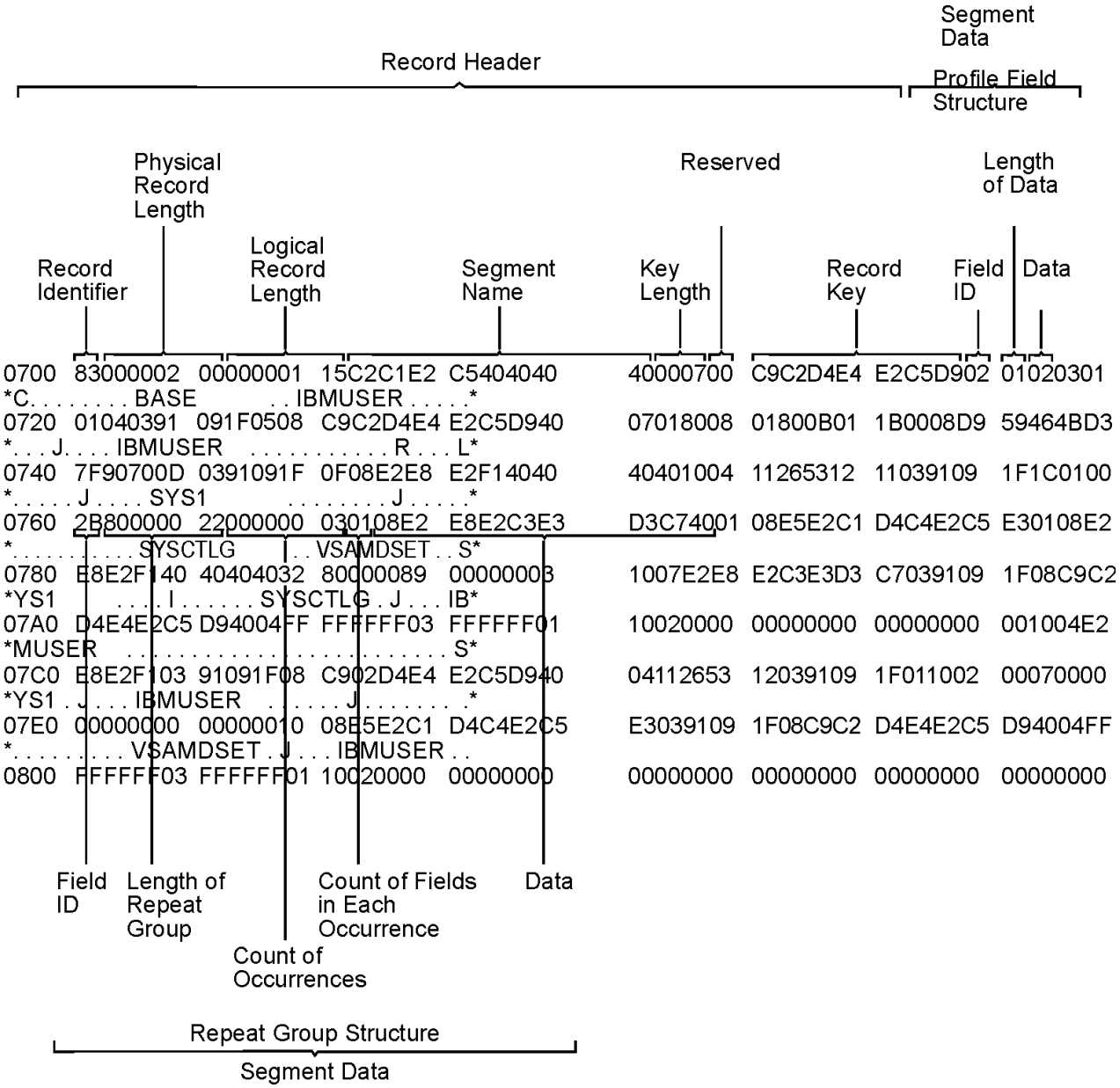


Figure 13. Example of a profile displayed by BLKUPD

Database Unload Utility for a RACF Database (IRRDBU00)

If you plan to use the block update (BLKUPD) command to update your database, you might want to get a better understanding of what your current database looks like.

The database unload utility (IRRDBU00) reads every profile as it unloads your RACF database to a sequential file. This can uncover problems in your database that you find only when reading the profile.

For more information on using the utility, see *z/OS Security Server RACF Macros and Interfaces* and *z/OS Security Server RACF Security Administrator's Guide*.

Note: RACF creates a member HWM for the SECDATA CATEGORY profile, which is reserved for IBM's use. The HWM member and a corresponding 0503 record exist if you added any categories to the SECDATA CATEGORY profile. Deleting the record impacts RACF behavior.

Block update command for a RACF database (BLKUPD)

You can use BLKUPD to examine or modify any block in a RACF database: a data block, index block, BAM block, or deallocated block.

To use BLKUPD, you must have at least UPDATE authority to the RACF database.

Attention:

Before using the BLKUPD command, you should be familiar with the RACF database and its configuration, because using BLKUPD improperly can damage the RACF database. (See “Format of the RACF database” on page 105.)

Before entering the BLKUPD command, you should read and understand the format of the database. Then, before using the BLKUPD command to perform updates to your RACF database, try to use one of the other RACF commands to alter or delete the entry in question.

This describes the following topics:

- Considerations when using the BLKUPD command
- Hierarchy and uses of the BLKUPD command
- Examples

Considerations for using BLKUPD

The following considerations apply to using the block update command (BLKUPD):

- In general:
  - Do not use BLKUPD during hours of peak activity on the active RACF database because the command uses hardware RESERVES or enqueues (GRS ENQs) to serialize access to the database.
  - BLKUPD can only be issued as a TSO command.
  - BLKUPD cannot be directed to run on another system using the AT or ONLYAT keywords or automatic command direction.
  - Whenever you need to run BLKUPD against a database that is active on a system that is a member of the RACF sysplex data sharing group, always run the utility from a system in the group, and make sure that you use the real name of the data set and not an alias. If you do not, you might damage your RACF database, or receive unpredictable results from the utility.
- Before using BLKUPD:
  - Try to correct the error by using other RACF commands.
  - Make a copy of the RACF database. Use BLKUPD on the copy. IBM might be able to use the copy for further diagnosis.
  - Run IRRUT200. The output might be useful in identifying the damaged portion of the database.
- Document all changes made for future reference and diagnosis.
- RVARY the "repaired" database online and test your fix.
- After the fix has been verified, apply the fix to the “real” database and RVARY this database ONLINE.
- If a profile is in question, delete the index entry that points to the profile, rather than trying to repair the profile itself. Use a RACF command to add the profile and index entry.
- Do not use BLKUPD to modify the BAM blocks. Instead, run the IRRUT400 utility, which rebuilds the BAM blocks.
- If a RACF database is RACF-protected, you must have at least UPDATE authority to the database.
- When executing BLKUPD under control of command procedures, you must add the DATA and ENDDATA command procedure statements around each BLKUPD command and subcommand that is also a command procedure statement. See *z/OS TSO/E Command Reference* for a description of command procedures.

- When executing BLKUPD under a TMP (terminal monitor program) that allows multitasking, you cannot have any other active task in your session. Allow the BLKUPD command to complete before executing any other TSO command.
- BLKUPD does not support attention handling.

  If an attention interrupt occurs and the user does not enter any data other than a null line or TIME command, the RACF database remains enqueued.
- The subcommands of BLKUPD follow the TSO syntax rules. See *z/OS TSO/E Command Reference* for a description of TSO command syntax rules.
- BLKUPD runs as an APF-authorized TSO command.
- TSO prompt must be set on.
- When RACF is enabled for sysplex communication, you cannot use BLKUPD to update the database if your system is operating in read-only mode.

If you need assistance in using BLKUPD, contact the IBM support center.

Hierarchy of the BLKUPD command

Figure 14 on page 121 shows the relationship between the BLKUPD command and its levels of subcommands.

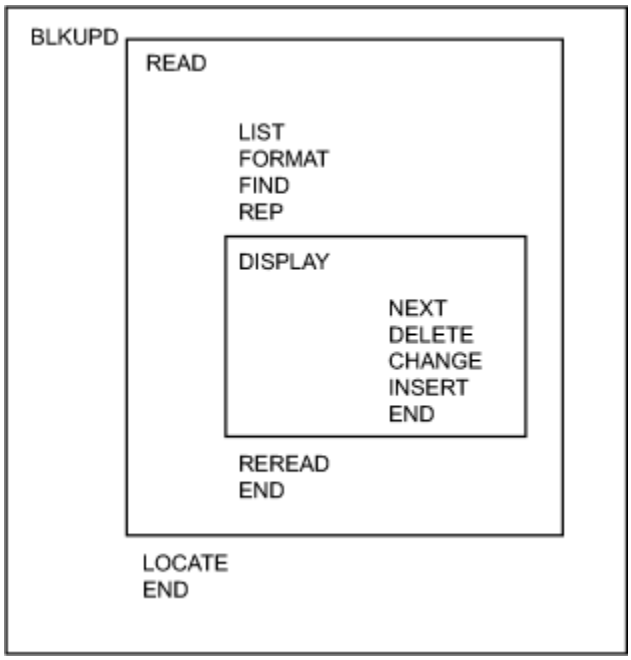


Figure 14. The relationship between the BLKUPD command and its levels of subcommands

Using the BLKUPD command

Use the BLKUPD command to access the RACF database. Do *not* use BLKUPD during hours of peak activity on the active RACF database because the command uses hardware RESERVES or enqueues (GRS ENQs) to serialize access to the database.

The Syntax of the BLKUPD command

```
BLKUPD [racfbname]
```

where *racfbname* specifies the name of a data set in the RACF database. The name can be from 1 to 44 characters.

If the name is not specified, the RACF database must be allocated to the TSO session with the SYSRACF file name.

Working with the RACF database

To update or examine a data set in the RACF database, perform the following steps:

1. Decide which data set in the RACF database you want to work with, and enter BLKUPD.

   Note: After entering the BLKUPD command, you can use the following subcommands: LOCATE, READ, or END.
2. Decide which block on the database you want to work with. If needed, use the LOCATE subcommand to assist you in finding the specific block.
3. Enter the READ subcommand, specifying either UPDATE or NOUPDATE.
4. Enter the subcommands of READ necessary to accomplish your task.
5. Issue the END command to end the utility.

Note that ALIAS index blocks are not supported by this utility. It is possible to use the LIST and the REP subcommands with ALIAS index blocks but great care should be taken since there is no format checking.

For more information about the alias indexes, see "Alias index blocks" on page 109.

The LOCATE subcommand

Use the LOCATE subcommand when you do not know the relative byte address (RBA) of the block you want to address. This subcommand locates an index entry in the sequence set and displays a formatted list of the contents of the level-1 index block containing the entry name. Optionally, the LOCATE subcommand displays a listing of all the index blocks in the chain from the highest-level index block to the appropriate level-1 block.

The syntax of the LOCATE subcommand (of BLKUPD):

```
LOCATE ENTRY(entryname)
       CLASS(classname)
       [GENERIC]
       [LISTALL]
```

where:

ENTRY(*entryname*)
Specifies the index entry to be located; *entryname* can be from 1 to 255 characters, which can be enclosed in quotation marks or not, and can contain hexadecimal data. For example:

```
ENTRY(I X'C2' MUSER)
```

is equivalent to:

```
ENTRY(IBMUSER)
```

To specify a mixed-case entry name, you must enclose *entryname* in quotations. If you do not enclose *entryname* in quotations, alphabetic characters in *entryname* are treated as uppercase characters, regardless of the case in which you enter them. For example, if you enter:

```
LOCATE ENTRY(irrsitec) CLASS(USER)
```

RACF looks for the index entry `IRRSITEC`, but if you enter:

```
LOCATE ENTRY('irrsitec') CLASS(USER)
```

RACF looks for the index entry `irrsitec`.

CLASS(*classname*)
is one of the following:

- GROUP
- USER
- DATASET
- The class name of the general resource

GENERIC
specifies that *entryname* is generic.

LISTALL
specifies that all the index blocks in the hierarchy chain are to be displayed.

The formatted list contains the same information as that produced by the FORMAT subcommand of READ (see "The FORMAT subcommand" on page 125).

If an index block in the chain contains an error, BLKUPD produces a hexadecimal dump of the block. If the name is not found in a level-1 block, BLKUPD displays the block that should contain it.

If the block containing the name is not found by a hierarchy index search because of errors in the chain, BLKUPD searches the sequence set.

The READ subcommand

The READ subcommand reads the block specified by the RBA (relative block address) into storage and copies it into a work area.

This subcommand allows you to examine or change the database.

The syntax of the READ subcommand (of BLKUPD):

```
READ rba [UPDATE]
         [NOUPDATE]
```

where:

rba
Specifies the relative byte address of the block to be read.

If the RBA does not represent an address on a 4KB boundary, the block on the 4KB boundary that contains the RBA is read. You can enter the RBA value as a hexadecimal (X'nn') or decimal (*nnn*) number. Hexadecimal numbers can be from 1 to 8 characters; decimal numbers can be from 1 to 10 characters.

UPDATE
Specifies you are planning to update one of the blocks in the RACF database.

Specifying the UPDATE keyword imposes an exclusive RESERVE on the database, meaning that the issuing system has exclusive control of the RACF database and only that system can update it. Any system that shares the database cannot read the database or update it. In addition, no other user on the issuing system can read or update the database.

If RACF is enabled for sysplex communication:

1. You cannot update the RACF database when you are operating in read-only mode.
2. When running in data sharing mode, RACF uses global resource serialization (ENQ/DEQ), rather than hardware RESERVEs.

NOUPDATE
Specifies you are examining one of the blocks in the RACF database.

Specifying the NOUPDATE keyword imposes a shared RESERVE on the RACF database, meaning that if you are sharing the RACF database with another system, both systems can read the database, but neither system can update it.

The NOUPDATE keyword on the READ command permits you to examine the database block without making any changes. NOUPDATE is the default. Should you make changes and then end BLKUPD, the changes are not recorded.

If RACF is enabled for sysplex communication and running in data sharing mode, RACF uses global resource serialization (ENQ/DEQ) rather than hardware RESERVEs.

Using the subcommands of READ

The UPDATE keyword on the READ subcommand allows you to use the subcommands to update the block. The subcommands are:

- **LIST**: Displays portions of the block in hexadecimal.
- **FORMAT**: Produces a formatted listing of the contents of an index block.
- **FIND**: Locates the offset of data in the block.
- **REP**: Replaces data in the block (valid only with the UPDATE command).
- **DISPLAY**: Displays the entries of an index block one at a time (it has several subcommands that are associated with it).
- **REREAD**: Overlays the new block in the work area with the old block (valid only with the UPDATE keyword).
- **END**: Terminates the READ function.

The FIND subcommand

The FIND subcommand finds the hexadecimal offset of a string in the block.

Syntax of the FIND subcommand of READ:

```
FIND string [OLD]
            [NEW]
```

where:

string
Specifies the string that is to be located.

The string can be hexadecimal (which is specified as X'*string*') or characters (which can be enclosed in quotation marks or not). Hexadecimal strings are right-aligned and padded with zeros to a byte boundary. Character strings are left-aligned and padded with zeros to a byte boundary.

If *string* contains lowercase characters, *string* must be enclosed in quotation marks. If *string* is not enclosed in quotation marks, all characters are treated as uppercase characters, regardless of the case in which you enter them.

OLD
Specifies the original block obtained by the READ command. OLD is the default.

NEW
Specifies the copy of the block. **NEW is valid only with the READ UPDATE function.**

After entering the FIND subcommand, enter the LIST subcommand to see the actual data.

The LIST subcommand

The LIST subcommand displays a hexadecimal dump of all or part of the RACF block. It allows you to see exactly what is inside the block and where it is located.

Syntax of the LIST subcommand of READ:

```
LIST [NEW|OLD]
     RANGE(xxx,yyy)
     ALL
```

where:

NEW
Specifies the copy of the original block (it might or might not have been updated). **NEW is valid only with the READ UPDATE function.**

OLD
Specifies the original block that was copied into storage when you entered the READ command. OLD is the default.

RANGE
Defines the parts of the block to be listed.

- *xxx* specifies the offset into the block where the listing is to start. The default is 0.
- *yyy* specifies the number of bytes to be listed. The default is 16.

The values for *xxx* and *yyy* can be hexadecimal or decimal numbers. If the sum of *xxx* and *yyy* exceeds 4096, the dump ends at the end of the block.

ALL
Specifies a hexadecimal listing of the entire RACF block.

The FORMAT subcommand

The FORMAT subcommand displays a formatted list of the contents of an index block. Use the FORMAT subcommand if you have read an index block and need to see the all the entries in this block and their RBAs (relative block addresses).

The list contains the following information:

- The RBA of the block
- The level of the block
- The offset to the last entry
- The offset to free space
- The offset of each entry within the block
- The front-end compression count of each entry
- The name of each entry (generic names are identified by a G in parentheses after the name)
- The last qualifier encode byte of each entry
- The RBA of the next-level index block for each entry or, for level-1 blocks, the RBA of the profile

If the block obtained by the READ command is not a valid index block, the request is rejected.

Syntax of the FORMAT subcommand of READ:

```
FORMAT [OLD|NEW]
```

where:

OLD
Specifies the original index block obtained by the READ command. OLD is the default.

NEW
Specifies the index block that was updated. **NEW is valid only with the READ UPDATE function.**

The REP subcommand

The REP subcommand replaces a string in the new block with another string. Optionally, this subcommand verifies the original string before making the replacement. **This subcommand is valid only with the READ UPDATE function.**

Syntax of the REP subcommand of READ UPDATE:

```
REP new-string
    OFFSET(xxx)
    [VER(old-string)]
```

where:

new-string
specifies the new string of data.

The data can be hexadecimal (which is specified as X'*string*') or characters (which can be enclosed in quotation marks or not). Hexadecimal strings are right-aligned and padded with zeros to a byte boundary. Character strings are left-aligned and padded with zeros to a byte boundary.

If *new-string* contains lowercase characters, *new-string* must be enclosed in quotation marks. If *new-string* is not enclosed in quotation marks, all characters are treated as uppercase characters, regardless of the case in which you enter them.

OFFSET(*xxx*)
Specifies the offset into the block where the string is to be replaced.

VER(*old-string*)
Specifies the original string that is to be verified. If the string is not found at offset *xxx* into the new block, the replacement is not made.

If *old-string* contains lowercase characters, *old-string* must be enclosed in quotation marks. If you do not enclose *old-string* in quotation marks, all characters are treated as uppercase characters, regardless of the case in which you enter them.

Note: The verify (VER) operand only verifies the characters specified; it does not verify the whole string. Be sure to enter the entire string you want verified.

The DISPLAY subcommand

The DISPLAY subcommand displays an entry of the index block that was read by the READ command.

When you enter the DISPLAY subcommand (after specifying READ UPDATE), the BLKUPD command copies the original block to a new block in storage. This copy is referred to as a **new** index block. You can then use the DELETE, CHANGE, and INSERT subcommands of the DISPLAY subcommand to modify the displayed entries.

If the block obtained by the READ command is not a valid index block, RACF rejects your DISPLAY request.

Syntax of the DISPLAY subcommand of READ:

```
DISPLAY [ENTRY(entryname) CLASS(classname)]
        [GENERIC]
```

where:

ENTRY(*entryname*)
Specifies the index entry where the display begins. If you do not specify an entry name, the display begins at the first entry in the index block; the entry name can be from 1 to 255 alphanumeric characters, which can be enclosed in quotation marks or not, and can contain hexadecimal data.

To specify a mixed-case entry name, you must enclose *entryname* in quotations. If you do not enclose *entryname* in quotations, alphabetic characters in *entryname* are treated as uppercase characters, regardless of the case in which you enter them.

If you specify an entry name, you must specify a class name.

CLASS(*classname*)

Must be one of the following:

- GROUP
- USER
- DATASET
- The class name of the general resource

GENERIC

Indicates that the index specified in ENTRY(*entryname*) is a generic name.

Using the Subcommands of DISPLAY

The subcommands of the DISPLAY subcommand are:

CHANGE
Changes the displayed entry (valid only with the READ UPDATE function).

INSERT
Inserts a new index entry immediately preceding the entry that is displayed (valid only with the READ UPDATE function).

NEXT
Displays the next entry.

DELETE
Deletes the displayed entry (valid only with the READ UPDATE function).

END
Ends the DISPLAY function.

The CHANGE subcommand of DISPLAY

`This subcommand is valid only with the READ UPDATE function.`

The CHANGE subcommand allows you to change the following:

- The name
- The RBA (relative byte address) of the index entry displayed
- The RBA of the segment specified
- The compression count (or the entry identifier byte or both)

The changed entry is displayed. Enter the NEXT subcommand or a null line to display the next entry.

Syntax of the CHANGE subcommand of DISPLAY:

```
CHANGE [ENTRY(entryname) CLASS(classname)]|[SEGMENT(segment-name)]
       [GENERIC]
       [RBA(rba)]
       [COMP(xx)]
       DUPLICATE|NODUPLICATE
```

where:

ENTRY(*entryname*)

Specifies the index entry where the display begins. If you do not specify an entry name, the display begins at the first entry in the index block; the entry name can be from 1 to 255 alphanumeric characters, which can be enclosed in quotation marks or not, and can contain hexadecimal data.

To specify a mixed-case entry name, you must enclose *entryname* in quotations. If you do not enclose *entryname* in quotations, alphabetic characters in *entryname* are treated as uppercase characters, regardless of the case in which you enter them.

If you specify an entry name, you must specify a class name.

CLASS(*classname*)

Must be one of the following:

- GROUP
- USER

- DATASET
- The class name of the general resource

The length of name plus the compression count cannot exceed 255. To create a fully compressed name, specify ENTRY (‘ ’), where the entry name is a null string, with a nonzero compression count. For more information about compression and an explanation of index blocks and encoding, see “Index entries” on page 111.

SEGMENT(*segment-name*)

Specifies the segment RBA to be changed.

The BASE segment cannot be specified as the segment name.

The ENTRY and CLASS keywords and the SEGMENT keyword are mutually exclusive. If both are specified, SEGMENT is ignored.

If you specify SEGMENT, only the RBA (relative byte address) of the SEGMENT is changed.

GENERIC

Indicates that the index entry name specified in ENTRY(*entryname*) is a generic name.

RBA(*rba*)

The value of *rba* can be from 1 to 8 hexadecimal characters or from 1 to 10 decimal characters.

COMP(*xx*)

The front-end compression count. *xx* is limited to one less than the length of the entry, and never exceeds 254. *xx* can be specified as decimal or hexadecimal characters.

DUPLICATE

Means that you want to make the entry a duplicate of the entry immediately following. This entry that immediately follows might be on the same block or appear at the beginning of the next block.

NODUPLICATE

Means that you do not want the current entry to be a duplicate of the entry immediately following.

There is no default on this keyword; if you do not specify DUPLICATE or NODUPLICATE, the entry identifier byte is not changed. It remains at its previous setting.

Specify only those keywords that you want to change. The others remain at their previous settings.

The BLKUPD command does not perform any checking to see if you have specified the appropriate keyword values to be changed. You must be very careful because specifying an incorrect keyword value or omitting a keyword and its value can produce unpredictable results.

The INSERT subcommand of DISPLAY

`This subcommand is valid only with the READ UPDATE function.`

The INSERT subcommand inserts a new index entry immediately preceding the entry that is displayed if SEGMENT is not specified. When SEGMENT is specified, the new segment ID is placed at the end of the segment array.

Note: The BLKUPD command cannot be used to insert the first entry in the database.

BLKUPD displays the new entry; entering the NEXT subcommand displays the original entry again.

Syntax of the INSERT subcommand of DISPLAY:

```
INSERT [ENTRY(entryname) CLASS(classname)]|[SEGMENT(segment-name)]
       [GENERIC]
       RBA(rba)
       [COMP(xx)]
       [DUPLICATE|NODUPLICATE]
```

where:

ENTRY(*entryname*)

Specifies the index entry where the display begins. If you do not specify an entry name, the display begins at the first entry in the index block. The entry name can be from 1 to 255 alphanumeric characters, which can be enclosed in quotation marks or not, and can contain hexadecimal data.

To specify a mixed-case entry name, you must enclose *entryname* in quotations. If you do not enclose *entryname* in quotations, alphabetic characters in *entryname* are treated as uppercase characters, regardless of the case in which you enter them.

If you specify an entry name, you must specify a class name.

CLASS(*classname*)

Must be one of the following:

- GROUP
- USER
- DATASET
- The class name of the general resource

The length of name plus the compression count cannot exceed 255. To create a fully compressed name, specify ENTRY (‘ ’), where the entry name is a null string, with a nonzero compression count. For more information about compression, see “Index entries” on page 111.

SEGMENT(*segment-name*)

Specifies the new segment to be inserted.

The BASE segment cannot be specified as the segment name.

The ENTRY and CLASS keywords and the SEGMENT keyword are mutually exclusive. If both are specified, SEGMENT is ignored.

GENERIC

Indicates that the index specified in ENTRY(*entryname*) is a generic name.

RBA(*rba*)

The value of *rba* can be from 1 to 8 hexadecimal characters or from 1 to 10 decimal characters.

COMP(*xx*)

The front-end compression count. *xx* is limited to one less than the length of the entry, and never exceeds 254. *xx* can be specified as decimal or hexadecimal characters.

DUPLICATE

Means that you want to make the entry a duplicate of the entry immediately following. This entry that immediately follows might be on the same block or appear at the beginning of the next block.

NODUPLICATE

Means that you do not want the current entry to be a duplicate of the entry immediately following.

The NEXT subcommand of DISPLAY

The NEXT subcommand displays the next entry.

Syntax of the NEXT subcommand of DISPLAY (READ):

```
NEXT
N
```

Entering a null line also displays the next entry.

The DELETE subcommand of DISPLAY

`This subcommand is valid only with the READ UPDATE function.`

The DELETE subcommand deletes the current index entry or segment, and the next entry is displayed.

Attention:

Before you change or delete the first entry in the index, or insert a new first entry, you must first decompress any compressed entries that follow in the index block.

The best way to decide which entries to decompress is to display the index block and then examine the names and their compression counts, paying attention to the second entry in the block. If the second entry is compressed, or does not match the first entry enough to allow for proper decompression of the following entries, you must first delete the first entry and then do the following to each entry that needs decompression:

- If the entry has a compression count, enter a CHANGE command:

  ```
  CHANGE COMP(0) ENTRY(xxxyyy) CLASS(classname)
  ```

  where *xxx* is the compressed front-end characters as indicated by the old compression count, and *yyy* is the name that the BLKUPD command gave you as the name of this entry. (Note that you need the GENERIC keyword on the CHANGE command if the entry is generic.)

  The second entry (now the new first entry) has now been changed.
- If this entry matches the old (first) entry for enough characters to decompress the remaining entries, you can stop. If it doesn't, you must decompress any entry that this new first entry does not match.
- To do so, enter NEXT commands until you get to the next entry that has to be changed. Then enter the CHANGE command to alter it as already described. Continue with the NEXT and CHANGE commands until all the entries in question have been resolved.

It might only be necessary to decompress that second entry; however, based on your examination of the listing, you might have to decompress all the entries. For an explanation of compression, see “Index entries” on page 111.

Syntax of the DELETE subcommand of DISPLAY:

```
DELETE    [SEGMENT(segmentname)]
D
```

where:

SEGMENT(*segment-name*)
deletes the specified segment from the current index entry.

The END subcommand of DISPLAY

This subcommand ends the DISPLAY subcommand.

You can continue to examine or update the block under the READ subcommand.

Syntax of the END subcommand of DISPLAY:

```
END     SAVE
        NOSAVE
```

where:

SAVE
Tells RACF to update the new index block to reflect the changes made under the DISPLAY subcommand. Specifying SAVE means that you want the new index block to replace the old one.

NOSAVE
Tells RACF to ignore all the changes you made under the DISPLAY subcommand with the UPDATE keyword specified on the READ subcommand.

If you did n4ot specify UPDATE on the READ command, then the NOSAVE option is forced.

The REREAD subcommand

The REREAD subcommand is valid only with the READ UPDATE function.

The REREAD subcommand allows you to overlay the new RACF block in the work area with the old block. Use this function when you have made errors to the new block while using the DISPLAY or REP subcommands. This subcommand gives you a new copy of the old block.

Syntax of the REREAD subcommand of READ UPDATE:

```
REREAD
```

The END subcommand of READ

The END subcommand ends processing on the RACF database block that you obtained by entering the READ command.

Syntax of the END subcommand of READ:

```
END     SAVE
        NOSAVE
```

where:

SAVE
Specifies that you want the new block to be written back to the RACF database to replace the old one that was read.

NOSAVE
Specifies that the RACF database not be updated with the new block. The NOSAVE option is forced if you did not specify the UPDATE keyword on the READ command or if you did not make any changes to the block.

If you want to use the READ function to examine or update another block, enter another subcommand.

The END subcommand of BLKUPD

The END subcommand ends BLKUPD.

The syntax of the END subcommand (of BLKUPD):

```
END
```

If you want to read another block, do not enter this final end statement (enter another subcommand of the BLKUPD command).

The BLKUPD command—A scenario

This example illustrates the functions of BLKUPD. It is not meant to be used for repairing your RACF database.

1. Issue the BLKUPD command on the RACF data set that you want to work with.

```
BLKUPD 'SYS1.RACF'
BLKUPD:
```

2. Replace a string in the block located at RBA X'E000'.

```
READ X'E000' UPDATE
BLKUPD:
```

reads the block at RBA X'E000' and specifies that changes are to be made to the block.

```
FORMAT
```

produces a formatted listing of the OLD index block obtained by the READ command.

```
BLOCK WITH RBA OF 00000000E000
OFFSET COMP.             ENTRY NAME                        RBA          BAM
       COUNT                                                        BLOCK BYTE BIT
 00E   000  BEATL                                    00000000DC00  00  02F   4
 02A   000  HILDE                                    00000000D000  00  02E   0
                          SEGMENT NAME: TSO          00000000D900  00  02F   1
                                        DFP          00000000DA00  00  02F   2
 051   000  HILDE.RDS.CNTL                           00000000DE00  00  02F   6
 073   000  IBMUSER                                  00000000D700  00  02E   7
 08E   000  M267                                     00000000DB00  00  02F   3
 0A6   000  M550ABC                                  00000000DD00  00  02F   5
 0C1   000  SALES                                    00000000DF00  00  02F   7
 0DA   000  SALES.* (G)                              00000000F000  00  032   0
 0F5   000  SECLABEL-SYSHIGH                         00000000D100  00  02E   1
 119   000  SECLABEL-SYSLOW                          00000000D200  00  02E   2
 13C   000  SECLABEL-SYSNONE                        00000000D300  00  02E   3
 160   000  SYSCTLG                                  00000000D600  00  02E   6
 17B   000  SYS1                                     00000000D400  00  02E   4
 193   000  VSAMDSET                                 00000000D500  00  02E   5
 1AF   000  255 X'FF'S
 2BB        SEQUENCE SET POINTER                    000000000000
TOTAL NAMES IN THIS BLOCK-015. UNUSED BYTES-3358. AVERAGE NAME LENGTH-025.
LEVEL NUMBER-01. DISPLACEMENT TO LAST KEY-02BB. DISPLACEMENT TO FREE SPACE-02C4
(G) - ENTITY NAME IS GENERIC
```

3. Find the string in the old block.

```
FIND M267
IRR63002I Offset is X'0000009A'
BLKUPD:
```

locates the offset of the character string ‘M267’ in the old block.

4. Verify that offset X'9A' in the new block contains the character string ‘M267’ and replace that string with the character string ‘M270’.

```
REP M270 OFFSET(X'9A') VER(M267)
 IRR63004I REPLACE complete.
```

5. List the modified area of the new block (8 bytes beginning at offset X'9A').

```
LIST NEW RANGE(X'9A',8)00000000E000
  009A D4F2F7F0 01010000
  *M270....                      *
```

6. Delete the first entry in the index block.

```
BLKUPD:
DISPLAY
 OFFSET COMP.       ENTRY NAME                                  RBA
        COUNT
```

```
  00E    000    SIVLE                                          00000000DC00
DELETE
```

deletes the index entry just displayed. The next entry, which moves up to replace the deleted one, is displayed.

```
IRR63036I The first index entry has been deleted. The rest of the index
block may need to be updated.
   00E    000    HILDE                                          00000000D000
                                        SEGMENT NAME: TSO   00000000D900
                                                      DFP   00000000DA00
```

7. Insert a new first index entry.

```
INSERT ENTRY(SIVLE) CLASS(USER) RBA(X'DC00')
```

inserts a new index entry, with the specified name, and RBA immediately preceding the entry just displayed. The new entry is displayed.

```
IRR63038I The first index entry has been inserted. The rest of the index
block may need to be updated.
   00E    000    SIVLE                                          00000000DC00
NEXT
```

displays the next index entry.

```
00E    000    HILDE                                          00000000D000
                                     SEGMENT NAME: TSO   00000000D900
                                                   DFP   00000000DA00
```

8. Change the name of the entry.

```
04C  000    HILDE.RDS.CNTL                                 00000000DE00
CHANGE ENTRY(HILDE.RDS.CHNG) CLASS(DATASET)
```

changes the entry name of the entry.

```
04C  000    HILDE.RDS.CHNG                                   00000000DE00
```

9. Save the changes you have made and obtain a listing of your updated block.

```
END SAVE
```

ends the DISPLAY command and saves the changes that were made in the index block. The utility still runs under the READ subcommand.

```
IRR63009I DISPLAY ended. Changes saved.
```

```
FORMAT NEW
```

produces a formatted listing of the updated index block.

```
BLOCK WITH RBA OF 00000000E000
OFFSET COMP.             ENTRY NAME                      RBA          BAM
       COUNT                                                    BLOCK BYTE BIT
 00E   000  SIVLE                                  00000000DC00  00  02F   4
 025   000  HILDE                                  00000000D000  00  02E   0
                          SEGMENT NAME: TSO       00000000D900  00  02F   1
                                        DFP       00000000DA00  00  02F   2
 04C   000  HILDE.RDS.CHNG                        00000000DE00  00  02F   6
 06E   000  IBMUSER                              00000000D700  00  02E   7
 089   000  M270                                  00000000DB00  00  02F   3
 0A1   000  M550ABC                               00000000DD00  00  02F   5
 0BC   000  SALES                                  00000000DF00  00  02F   7
 0D5   000  SALES.* (G)                           00000000F000  00  032   0
 0F0   000  SECLABEL-SYSHIGH                      00000000D100  00  02E   1
 114   000  SECLABEL-SYSLOW                       00000000D200  00  02E   2
 137   000  SECLABEL-SYSNONE                      00000000D300  00  02E   3
 15B   000  SYSCTLG                               00000000D600  00  02E   6
 176   000  SYS1                                  00000000D400  00  02E   4
 18E   000  VSAMDSET                              00000000D500  00  02E   5
```

```
 1AA   000  255 X'FF'S
 2B6        SEQUENCE SET POINTER                    000000000000
TOTAL NAMES IN THIS BLOCK-015. UNUSED BYTES-3363. AVERAGE NAME LENGTH-024.
LEVEL NUMBER-01. DISPLACEMENT TO LAST KEY-02B6. DISPLACEMENT TO FREE SPACE-02BF
(G) - ENTITY NAME IS GENERIC
```

10. Save the changes made and write them back to the RACF database.

```
END SAVE
IRR63013I READ ended. Block saved.
```

    ends the READ command and writes the new block out to the RACF database.

11. Locate the block that contains the generic name of SALES.*.

```
LOCATE ENTRY(SALES.*) CLASS(DATASET) GENERIC
```

    displays a formatted list of the level one block that contains the generic entry name ‘SALES.*’.

```
BLOCK WITH RBA OF 00000000E000
OFFSET COMP.             ENTRY NAME                      RBA         BAM
       COUNT                                                   BLOCK BYTE BIT
 00E   000  SIVLE                                  00000000DC00  00  02F   4
 025   000  HILDE                                  00000000D000  00  02E   0
                           SEGMENT NAME: TSO       00000000D900  00  02F   1
                                         DFP       00000000DA00  00  02F   2
 04C   000  HILDE.RDS.CHNG                        00000000DE00  00  02F   6
 06E   000  IBMUSER                               00000000D700  00  02E   7
 089   000  M270                                  00000000DB00  00  02F   3
 0A1   000  M550ABC                               00000000DD00  00  02F   5
 0BC   000  SALES                                  00000000DF00  00  02F   7
 0D5   000  SALES.* (G)                           00000000F000  00  032   0
 0F0   000  SECLABEL-SYSHIGH                      00000000D100  00  02E   1
 114   000  SECLABEL-SYSLOW                       00000000D200  00  02E   2
 137   000  SECLABEL-SYSNONE                      00000000D300  00  02E   3
 15B   000  SYSCTLG                               00000000D600  00  02E   6
 176   000  SYS1                                  00000000D400  00  02E   4
 18E   000  VSAMDSET                              00000000D500  00  02E   5
 1AA   000  255 X'FF'S
 2B6        SEQUENCE SET POINTER                  000000000000
TOTAL NAMES IN THIS BLOCK-015. UNUSED BYTES-3363. AVERAGE NAME LENGTH-024.
LEVEL NUMBER-01. DISPLACEMENT TO LAST KEY-02B6. DISPLACEMENT TO FREE SPACE-02BF
(G) - ENTITY NAME IS GENERIC
```

12. End the BLKUPD session.

```
BLKUPD:
END
READY
```

    terminates the BLKUPD command.

Chapter 5. Error recovery for RACF remote sharing facility (RRSF)

This topic describes:

- The flow of a directed command in a RACF remote sharing facility (RRSF) environment
- The types of errors that are experienced in an RRSF environment
- Handshaking between RRSF nodes
- The connection states of RRSF nodes and the transitions between them
- Actions to recover from an RRSF failure
- The recording of RRSF errors

RRSF moves the RACF command between the user who initiated the command and the RACF subsystem address space where the transaction is processed (either on a local or a remote node) and moves the RACF command output between the RACF subsystem address space where the transaction was processed (on a local or a remote node) and the node where the user was logged on when the transaction was initiated.

The INMSG data set is used to temporarily hold requests that are being sent to the local node from itself or a remote node, such as commands directed to the local node or output from RACF commands that were directed to a remote node. The OUTMSG data set is used to temporarily hold requests that are being sent to a target node, such as commands directed from the local node. If RRSF experiences a failure, it:

- Detects the error when it first occurs. This:
  - Prevents additional damage to the RACF subsystem address space and the RACF database.
  - Allows either the system programmer or the IBM support center to diagnose and fix the problem.
- Protects the rest of the RRSF network from damage by isolating the error to the failing node.
- Saves the work request for retry when the error has been corrected.

Figure 15 on page 136 contains an illustration of the flow of a directed command in an RRSF environment. The steps that are shown in this figure are:

1. A user issues a RACF command. This command enters the local node's OUTMSG data set while waiting to be sent to the remote node.
2. When this command is sent to the remote node to be processed, it enters the remote node's INMSG data set.
3. The command is processed in the remote node's RACF subsystem address space and the results are placed in the remote node's OUTMSG data set while waiting to be sent to the local node.
4. The results are sent to the local node and placed in the local node's INMSG data set until the output is returned to the user.

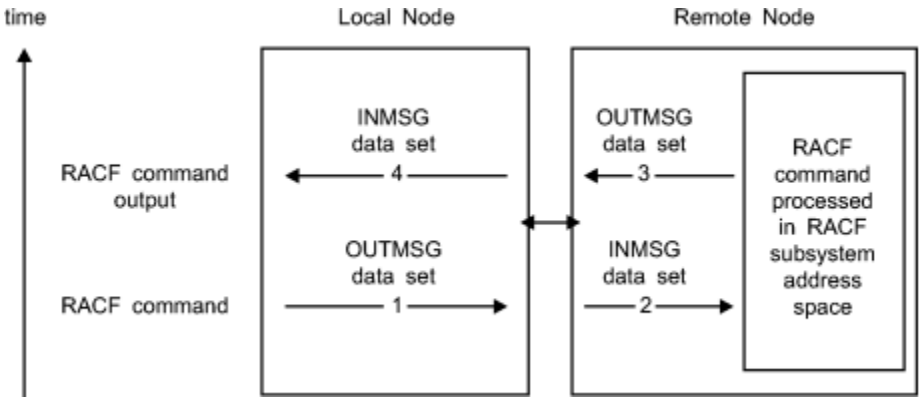


Figure 15. Flow of a directed command in an RRSF environment

Types of errors in an RRSF environment

The following types of errors can be experienced by an RRSF network:

Definition errors

For RRSF, typical definition errors are local or remote nodes not being correctly defined to the system or to each other. In general, definitions must be accurate and consistent on both the local and target node. Problems occur when there are:

- For one of the following:
  - Previous versions of RACF database templates
  - Inconsistent node definitions to RACF (that is, something that is specified on the TARGET command at a local node was specified differently on a corresponding TARGET command at a remote node)
  - Incorrect security information for VSAM data sets
  - Differences in node status (NODEA is operative while NODEB thinks NODEA is dormant).
- For APPC:
  - Incorrect VTAM® definitions
  - Incorrect LU names
  - Incorrect security information for APPC conversation
- For TCP/IP:
  - Incorrect TCP/IP definitions
  - Incorrect host address names
  - Incorrect security information for TCP/IP
  - AT-TLS policy problems
  - Digital certificate and key ring problems

Environmental errors

Typical environmental errors are MVS resources not being available. These can occur when MVS or RACF is being restarted or can occur because of high volumes of directed commands (including automatically directed commands) or password changes that are being synchronized. This type of error can occur on either the local or remote node, and includes:

- For one of the following:
  - Insufficient main memory
  - Insufficient VSAM space

  - Remote RACF subsystem address space not active
- For APPC:
  - VTAM not yet active
  - APPC address space not yet active
  - Retryable APPC errors
- For TCP/IP:
  - TCP/IP policy agent or resolver address spaces not yet active
  - Retryable TCP/IP errors
  - Network connectivity problems
  - Firewall issues
  - Name server problems

Program errors

Overlaid control blocks, overlaid in-storage data, or programming errors within RRSF, and the portions of RACF started within the RACF subsystem address space can also cause an abend. Non-fatal errors detected by RRSF can have the following symptoms:

- Units of work being lost
- Units of work being done multiple times
- All tasks in the RACF subsystem address space being in wait state
- Resources are allocated to a request and the request has completed either normally or abnormally and the resources are not released or deallocated

Transmission and line errors

Transmission and line errors are external to the application, such as a car hitting a telephone pole.

Handshaking and state changes

Handshaking involves verifying that two RRSF nodes can communicate with each other and are compatible. Handshaking occurs whenever a TARGET command is issued to activate the connection between the local node and a remote node if sufficient information has been provided. Activating a remote node connection requires sufficient information to do all of the following:

- Allocate and open the VSAM data sets (INMSG and OUTMSG data sets)
- Register the RACF subsystem address space (as an APPC/MVS server for the APPC protocol, or a socket listener for the TCP/IP protocol)
- Define the required protocol information for the remote node

Connection states of RRSF nodes

While at a high level there are two types of connections between nodes, operative and dormant, at a lower level the connection between two nodes can be in any one of a number of states.

The state of the connection between nodes and the transitions between the states that are used by RRSF have the following characteristics:

- Controls whether information is sent to the remote node.
- Respects the state change request of the last person who issued such a request; that is, the connection status is not changed from dormant to active without an explicit human action.
- Provides available information about how the connection reached the current state.

- Recognizes that certain state changes do not occur immediately because they must check with the partner node to determine if the new state is allowed.

The following table shows the potential states that can exist on a local node to describe the connection between that node and a remote node:

Table 30. Connection states between nodes

Name	Abbreviation	Description
operative pending connection	O-P-C	The local node has requested that the connection be activated and is attempting to activate the conversation. The local node has not yet received a confirmation that the remote node will accept the connection. This state can occur while the local node is attempting to restore a successful connection that was interrupted.
operative pending verification	O-P-V	The local node's request for a conversation has been accepted. The two nodes are communicating and evaluating information they have exchanged to determine if they are compatible. The information can include RACF information and the digital certificates exchanged during the TLS handshake performed by TCP/IP. If the two nodes are not compatible, both nodes will remain in the operative pending verification state.
operative active	O-A	The connection between two nodes is active. The two nodes have verified that they can communicate with each other and that they are compatible with each other.
operative in error	O-E	A non-retryable error has occurred related to workspace data sets (for example, a volume has gone offline). Console or syslog messages have been issued with more information.
dormant by local request	D-L	The local node's connection with a remote node has been made dormant by an operator issuing a TARGET DORMANT command.
dormant by remote request	D-R	The local node has detected that the connection to the remote node has been made dormant by the remote node or the APPC connection between the local and remote nodes has not been defined on the remote node. This state can occur if the local node cannot successfully connect with a node after a reasonable amount of time.
dormant by mutual request	D-B	The local and remote nodes have both requested the connection be dormant by an operator issuing a TARGET DORMANT command on each system.

Table 30. Connection states between nodes (continued)

Name	Abbreviation	Description
dormant in error	D-E	The local node is dormant and a failure is experienced while saving RRSF requests for later processing.
defined	DEF	TARGET information has been defined, but no conversation occurs. This state occurs: • Between member systems of a multisystem node. Systems in a multisystem node do not communicate with each other. • Between a local nonmain system and a nonmain system on a remote multisystem node. Nonmain systems of multisystem nodes can communicate with single-system nodes and with the main systems of multisystem nodes, but they do not communicate with nonmain systems of other multisystem nodes.
not defined (initial)	???	No connection has been established to the node due to insufficient configuration information, or because a TARGET OPERATIVE or TARGET DORMANT command has not been issued for the node. The TARGET LIST command output shows this state as ???.

These connection states describe the state of the local node and the state of a connection between the local node and a remote node.

For example, from the local node perspective, operative active (O–A) means that the local node can have either a TCP/IP or APPC listener process and accepts connections from remote nodes. Dormant by local request (D–L) means that there are no listener processes active and no new conversations are accepted (existing O–A conversations are unaffected).

These connection states also describe the state of the local node when you have no remote node connection because you have not specified protocol information. The local node can be operative active (O–A), but does not have active listener processes. The local node can be dormant by local request (D–L), but not dormant by remote request (D–R) or dormant by mutual request (D–B). If VSAM errors occur, the local node can be either operative in error (O–E) or dormant in error (D–E).

Connection state changes after a successful TARGET command

The following table shows the connection state as seen on the local node after a successful TARGET command is issued on the local node to change the connection state between the local and remote nodes.

For example, if the connection state as seen on the local node is operative in error (O–E) and the local node issues a successful TARGET DORMANT command, the connection state, as seen by the local node, would be dormant by local request (D–L).

Table 31. Table of new connection states based on a successful TARGET command

Previous connection state	After a successful TARGET OPERATIVE command	After a successful TARGET DORMANT command	After a successful TARGET DELETE command
O–A	O–P–C	D–L	error message received

Table 31. Table of new connection states based on a successful TARGET command (continued)

Previous connection state	After a successful TARGET OPERATIVE command	After a successful TARGET DORMANT command	After a successful TARGET DELETE command
O–P–C	O–P–C	D–L	error message received
O–P–V	O–P–C	D–L	error message received
O–E	O–P–C	D–L	error message received
D–L	O–P–C	D–L	connection deleted
D–R	O–P–C	D–B	error message received
D–B	O–P–C	D–B	connection deleted
D–E	O–P–C if INMSG and OUTMSG workspace data sets can be opened, else D–E	D–L if INMSG and OUTMSG workspace data sets can be opened, else D–E	connection deleted
DEF	DEF Message IRRM035I is issued to syslog.	DEF Message IRRM035I is issued to syslog.	connection deleted

Actions to recover from an RRSF failure

RRSF can experience the following types of failures:

1. An ABEND that causes ESTAE processing. For detailed information, see "RRSF ESTAE processing" on page 140.
2. An error return code from an MVS system service. For detailed information, see "Failure of MVS system service" on page 141.
3. An error return code from APPC/MVS. For detailed information, see "APPC return code handling" on page 147.
4. An error return code from a z/OS UNIX System Services socket API. For detailed information, see "TCP/IP return code handling" on page 144.
5. An error return code from a request to VSAM. For detailed information, see "VSAM failures" on page 141.

RRSF ESTAE processing

ESTAE or ESTAI processing occurs when MVS detects an abnormal end during processing of a task within the RACF subsystem address space.

The actions indicated here are normally done internally by RACF itself. However, in some cases, a manual RESTART command might be necessary.

The following is done during recovery processing:

1. If the task that failed had not abended during recovery processing:
   a. If the task is initializing:
      i) The task terminates to clean up its resources.
      ii) Another attempt is made to re-create the task.
   b. If the task is processing an item:
      i) The queue of waiting work is preserved.
      ii) The current task is marked as being in recovery processing.

iii) In-use resources for the current item are freed.

iv) The current item is restarted.

c. If the task is terminating, termination continues and the abend is ignored.

d. If the task that failed is in recovery processing:

i) Resources in use for the task are released.

ii) If the failure occurs when processing the same item for the second time, the item is deleted and output to be returned to the originator is created.

iii) The task that has failed is restarted.

Failure of MVS system service

When an error return code from an MVS system service is detected, an attempt is made to recover the work being done by RRSF. The basic retry scheme depends on where the error is detected. Most of the tasks within RRSF have the following structure:

1. An initialization section
2. A processing loop
3. A termination section

If errors are detected in the initialization section, the error is recorded and the task is restarted.

Note: The RACF subsystem address space attempts to restart a task *five* times before an abend is issued.

The actions indicated here are normally done internally by RRSF itself. However, in some cases, a manual RESTART command might be necessary.

If errors are detected within the processing loop, the following steps are done:

1. An entry is created in the LOGREC data set at the time of failure.
2. The element is retried.
3. If retry fails,

   a. Another entry is created in the LOGREC data set.

   b. The failing item is discarded.

   c. If possible, returned output is sent to the originator.

   d. The task is restarted.

If errors are detected in the termination section, the error is recorded and termination continues. The termination code is usually entered when the task in the RACF subsystem address space is being restarted or stopped.

VSAM failures

The checkpoint information is stored in a VSAM key-sequenced data set (KSDS). These files can be unusable for a number of operational reasons, such as:

- There are I/O errors on the device that contains the VSAM data set.
- The operator has made the device inactive or unavailable.
- There is insufficient space to store the next record.

The actions indicated here are normally done internally by RRSF itself. However, in some cases, a manual RESTART command might be necessary.

If VSAM errors occur on the INMSG or OUTMSG data set, that prevent the records from being written to or deleted from the data sets, RRSF does the following:

- Shuts down the connection by:

  - Rejecting all inbound transactions by marking all affected successfully established connections as operative in error
  - Marking all dormant connections as dormant in error
  - Rejecting all directed commands that would require the use of the damaged VSAM data set
- Notifies the appropriate people by:
  - Writing a message to the system console
  - Creating a symptom record in the LOGREC data set
- Allows for correction by:
  - Attempting to close and de-allocate the VSAM data set that is experiencing errors

An error message is issued to the MVS operator console when a VSAM data set experiences an error. Some common error messages are IRRC029I and IRRC030I, which indicate problems in trying to write to the workspace data sets, and IRRC031I, which indicates that a workspace data set is full. Associated messages IRRC032I and IRRC033I are also issued, indicating that the specified connection has received a DORMANT ERROR or OPERATIVE ERROR.

After the error has been corrected, the following commands must be processed to reactivate the connection:

```
TARGET NODE(nodename) DORMANT
TARGET NODE(nodename) WORKSPACE([current or new settings])
TARGET NODE(nodename) OPERATIVE
```

Note: You should use RACF profiles to protect the INMSG and OUTMSG data sets.

Automatic direction and VSAM errors

Automatic direction is not apparent to most users who issue RACF commands, change their passwords, or take actions that result in application updates. They are typically not aware that their actions have caused an RRSF request to be automatically directed. Therefore, VSAM errors are not reported to the issuer of the request. Instead, the users who are specified on the NOTIFY and OUTPUT keywords of the RACF SET command (which might include the command issuer using &RACUID) receive an error message that the RRSF request was not sent to the appropriate remote node because of a VSAM file failure.

VSAM failures are critical because the request and the returned output are checkpointed in the VSAM files. If the VSAM files are unusable, there is no record of the RACF request or the output.

When an RRSF request generated by automatic direction is processed and it experiences a hard VSAM failure (that is, it cannot be written to the file and has not already been saved somewhere else) the standard failure output is sent to the users who are specified on the NOTIFY and OUTPUT keywords of the RACF SET command. This hard VSAM failure can only occur on the node where the request originated and before it is stored in the OUTMSG data set for the target node. In all other cases of a VSAM failure, a copy of the request or output exists in another VSAM file.

Password synchronization and VSAM failures

If possible, VSAM failures for password synchronization requests are processed the same as a VSAM failure for a directed command. A password synchronization failure generates RRSFLIST output and an error message to the TSO user.

IRRBRW00 utility and VSAM data sets

If you have a DASD problem or have a need for offline diagnosis of the RRSF VSAM data sets, RACF has an APF-authorized utility called IRRBRW00 that allows you to dump the RACF data contained within the VSAM data sets. Because this utility provides the capability to dump sensitive RACF data such as RACF command text and returned output, be careful to control access to this utility.

To prevent inadvertent disclosure of the data in the VSAM data sets, RACF requires that a user obtain read access to the resource name IRRBRW00 in the RRSFDATA class so the IRRBRW00 utility can be run. If the

class is not active or no profile is defined, the user is denied access to run the utility. The user also needs RACF access to the VSAM data set to be dumped. VSAM data sets can be protected by using profiles in the DATASET class. See Workspace data sets for more information about data set names.

To ensure the integrity of the data in these files, set up these profiles and limit the access to only those with a need. You can use the RACF TARGET command to determine the VSAM data set names being used by the RACF remote sharing facility.

The IRRBRW00 utility is provided for diagnostic purposes only. For complete instructions on running IRRBRW00, see the RACJCL member of SYS1.SAMPLIB and read the IRRBRW00 sample.

Node communications

TCP/IP and APPC are communication vehicles for sending and receiving messages from one RRSF node to another.

TCP/IP errors

This section contains information about:

- Connection processing during system IPL
- TCP/IP return code handling
- Network connectivity issues
- AT-TLS errors
- Obtaining information about RRSF connections

Note: Many network-related problems are general in nature, and are not specific to RRSF. When experiencing connectivity errors, use diagnostic tools provided by the Communication Server, not by RACF. See *z/OS Communications Server: IP Diagnosis Guide* for information about debugging network problems.

Connection processing during system IPL

During system IPL, the RACF subsystem is generally started earlier than some of the TCP/IP-related address spaces on which the RRSF TCP protocol support relies. For example, the TCP/IP and policy agent address spaces are required. Further, if host names are used instead of IP addresses in TARGET command definitions, then the resolver address space, and possibly an external name server, are also required. During RACF subsystem initialization, the socket listener process starts when the local node is made operative. Remote TCP connections are not attempted until the listener is successfully established. If the listener fails to initialize, message IRRC050I is issued and the listener periodically tries again for approximately 30 minutes, then issues message IRRC063I and stops trying.

As the TCP/IP address space initializes, it receives AT-TLS policy files from the policy agent address space. Address spaces (such as the RACF subsystem) might be requesting socket services before TCP/IP has obtained its policy files. The default behavior of TCP/IP is to reject such a request unless the user ID has access to a stack initialization resource that is used as an override. If this resource is protected by a RACF profile, then failed accesses result in ICH408I messages to the console. Therefore, you might see such a message for the RACF address space each time it attempts to establish its TCP listener before TCP/IP obtains the policy files. This is normal, and such messages can be ignored. Make sure that you do not allow the RACF subsystem user ID to this resource, or else the listener successfully initializes, and then attempts to establish connections with remote nodes. If these connections are attempted before TCP/IP obtains its policy, RRSF rejects the connections for lack of AT-TLS policy, and the connections must be restarted manually when the policy server has served the policy files to TCP/IP.

See *z/OS Communications Server: IP Configuration Guide* for information about stack initialization protection.

TCP/IP return code handling

RRSF uses z/OS UNIX System Services APIs to start socket services. Whenever an error is encountered, a message is issued to the console. The message contains the service name, the return and reason code from the service, and an additional diagnostic code that helps IBM service, if contacted. See *z/OS UNIX System Services Messages and Codes* for the return and reason codes. The return code is referred to as the *errno*, and the reason code is referred to as the *errnojr*. When looking up the reason code, use only the low-order halfword of the displayed value. In *z/OS UNIX System Services Messages and Codes*, there is a name and a value for each return code. Information about the identified service in *z/OS UNIX System Services Programming: Assembler Callable Services Reference* often lists some common errors by name and possible causes.

If the TCP socket listener encounters an error during initialization, or if a remote connection cannot be established, a console message is issued, and the attempt is periodically tried again. RRSF tracks the failing service name, including its return and reason code, and does not issue an error message if encountering the same error on a subsequent attempt.

Certain errors are not considered unrecoverable, and in such cases, RRSF does not issue an error message, but creates a LOGREC entry. These entries might be of value to IBM service when debugging a problem. See "Creation of LOGREC data set entries" on page 148 for more information.

Network connectivity issues

You are most likely using Communication Server tools and diagnostics to debug network problems. Therefore, you need the IP address of the system that the local system is attempting to communicate. Because the TARGET command allows you to specify the host address as a host name, it might not be clear which IP address this name corresponds to. Whenever RRSF resolves a host name to an IP address, it saves the address so it can be displayed by the TARGET LIST command. Note that for the local node, the IP address is resolved when a remote connection is attempted, not when the listener is established. If you use the default address (0.0.0.0) for the local node and want to know the resolved IP address, and TARGET LIST is not already displaying it, you should attempt to establish a remote connection, and then issue the TARGET LIST for the local node again.

AT-TLS errors

While establishing a connection with a remote RRSF node, the initiating system issues a socket connect() call and the receiving system issues an accept(). Then, both sides of the connection issue a select() on the established socket. The select causes TCP/IP to perform the TLS handshake with the assistance of System SSL. The handshake process determines the AT-TLS policy rule used to protect the connection on each system and applies that policy. The policy identifies the RACF key ring that contains the digital certificates required to authenticate each server to the other.

If the handshake fails, RRSF usually issues message IRRI031I. AT-TLS tracing, by default, logs errors and provides an error code. See *z/OS Communications Server: IP Diagnosis Guide* for more information about the error code. The AT-TLS trace level is specified in the AT-TLS rule for a connection. Because the TLS handshake is performed on both systems and the error might have occurred on only one of the systems, be sure to look in the trace log on the remote system if there is no helpful information about the system you are currently logged on to.

TLS handshake errors are usually caused by certificate or key ring setup errors, and this might not always be obvious from the error code description. The following checks generally identify the problem:

- On each system, use the RACDCERT LISTRING command to check that the key ring specified in the policy rule (the sample policy that RACF provides in the IRRSRRSF member of SYS1.SAMPLIB specifies IRR.RRSF.KEYRING and if using IBM Configuration Assistant, it specifies tlsKeyring) is defined for the RACF subsystem user ID. Note that key ring names are case-sensitive.
- Verify that the key ring contains a digital certificate for the RRSF server (the RACF subsystem user ID) as the default.
- Verify that the key ring contains the RRSF signing certificate and it is trusted.

- Verify that the RACF subsystem user ID has authority to read its own key rings (this is generally accomplished by granting READ access to the IRR.DIGTCERT.LISTRING resource in the FACILITY class). Note that this permission is required even if the RACF subsystem started task definition specifies TRUSTED or PRIVILEGED.

Other problems that might occur are:

- A certificate has expired.
- The key type associated with the certificate is not valid for the cipher algorithm requested in the policy.
- Some aspect of the AT-TLS policy requires ICSF, (for example, if the certificate private key is stored in the ICSF PKDS), but ICSF was not started when RRSF connections were attempted.
- There is a logical inconsistency between the policy files as they exist on the local and the remote system. Typically, both systems should have the same policy statements for RRSF, both for the "client" and "server" portions of the policy.

After correcting any key ring problems, including an authorization problem, make sure that the policy agent reads the contents of the key ring again, by changing the EnvironmentUserInstance value in the policy rule. If you are using Configuration Assistant, click Reaccess Key Rings... under image level settings, for a given image. After the policy is updated, refresh the policy agent by issuing the following command from the console:

```
F PAGENT,UPDATE
```

After correcting an ICSF problem (such as starting ICSF after an RRSF connection has failed because ICSF was not available), you must change the GroupUserInstance value, and then update the policy agent as shown above. Your system configuration must start ICSF earlier in the IPL sequence (before the policy agent starts) so that you avoid this problem on the next IPL.

Note that these keywords might not be in your policy. Specifically, if using the sample policy, the GroupUserInstance keyword is not specified. If so, see for information about where to add this statement in your policy.

Obtaining information about RRSF connections

The z/OS Communications Server NETSTAT command is frequently used to display information about TCP/IP connections. On a busy system, the output can be voluminous. To help identify RRSF sockets, RACF associates application data with each of the sockets it uses. Each socket is tagged with the string "IRRRRSF" to identify it as an RRSF socket. You can issue the NETSTAT command specifying application data to restrict the output to RRSF connections. For example:

```
NETSTAT ALLCON (APPLD IRRRRSF
```

The NETSTAT command displays output such as:

```
EZZ2350I MVS TCP/IP NETSTAT CS V1R13 TCPIP NAME: TCPIP 13:33:07
EZZ2585I USER ID CONN LOCAL SOCKET FOREIGN SOCKET STATE
EZZ2586I ------- ---- ------------ -------------- -----
EZZ2587I RSWJ 0000000A 9.57.1.243..18136 0.0.0.0..0 LISTEN
EZZ2591I APPLICATION DATA: IRRRRSF
EZZ2587I RSWJ 000000A3 9.57.1.243..18136 9.57.1.13..1034 ESTABL
SH
EZZ2591I APPLICATION DATA: IRRRRSF
```

Figure 16. NETSTAT command output

This output shows two RRSF sockets. The first is for the listener process, and the second is for a remote node connection. The output shows that the remote node connection was initiated by the remote node, because the local socket value is the default listener port of 18136, and the remote socket uses an ephemeral port number of 1034.

Obtaining AT-TLS information for an RRSF connection

TARGET LIST displays some basic information extracted from the AT-TLS rule covering a remote connection, such as:

- The rule name
- The negotiated cipher
- The client authentication level
- If the client authentication level is SAFCHECK, the local user ID that the remote server's certificate mapped

This information can be helpful as a quick check, however, the NETSTAT command can be used to obtain all the policy information.

See Figure 16 on page 145 for an example of NETSTAT command output and note that a connection identifier is shown in the CONN column. For the remote node, the connection ID shown is A3. To obtain detailed information about the AT-TLS policy covering that connection, issue the NETSTAT command specifying the connection ID:

```
NETSTAT TTLS CONN A3
```

or

```
NETSTAT TTLS CONN A3 DETAIL
```

See *z/OS Communications Server: IP System Administrator's Commands* for more information about the NETSTAT command.

APPC errors

This section contains information about:

- Initializing connection errors
- APPC return code handling

Initializing of connections

There is a class of problems that occur because of the order that the MVS address spaces become activated during a system IPL. As an MVS system is being IPLed, address spaces reach a state in which they are able to service requests at different rates. Therefore, there is no assurance that the VTAM and APPC address spaces, that are necessary to service RRSF requests, are available when the RACF subsystem address space issues its first request.

Attempting to activate the connections or send information to remote nodes when the RACF subsystem address space is started, but before VTAM and APPC have completed initialization, result in APPC failures. Normally, such APPC failures cause the connections between the two nodes to become operative in error when the first attempt is made. Instead, RRSF allows the RACF subsystem address space to continue to poll for APPC/MVS availability and when available, attempts to register the RACF subsystem address space as an APPC/MVS server.

If the RRSF APPC server fails to successfully register with APPC/MVS, error message IRRC023I is issued to the MVS operator console and RRSF initialization discontinues.

A registration attempt to APPC/MVS also occurs if:

- A local node is in a dormant state,
- A TARGET command is issued to make the connection operative, and
- Protocol information has been supplied.

A subsequent TARGET LIST command against the node would show a connection state of operative active.

Note: If MODENAME is not specified in the TARGET command, RACF sends a default name to APPC that allows the connection to complete successfully. This default name, IRRMODE, is placed in IRRNODE. However, since there was no explicit specification by the user and the default name is a RACF internal, the TARGET LIST output shows "NOT SPECIFIED".

APPC return code handling

The actions indicated here are normally done internally by RRSF itself. However, in some cases, a manual RESTART command might be necessary.

The MVS callable system services, that are used to implement the APPC or LU6.2 protocols, provide return codes to report the status of the request. Normally, a failure causes:

- The connection to be marked as operative in error
- A symptom record to be created in the LOGREC data set. See "Creation of LOGREC data set entries" on page 148 for details.

However, if RACF receives one of the following three error conditions back from APPC, RACF tries the error again:

- Allocate_failure_retry
- TP_not_available_retry
- Resource_failure_retry

If after 10 times, RACF continues to receive an error return code, it is considered a non-retryable error and:

- The connection is marked as operative in error.
- A symptom record is created in the LOGREC data set. See "Creation of LOGREC data set entries" on page 148 for details.

APPC missing interrupts

Another type of APPC error occurs when the MVS callable system service that attempts to perform an APPC function never completes. One way this can occur is if the partner remote RACF subsystem address space starts the TP program, but does not complete the request. A missing interrupt handler (MIH) uses a timer to wait for the transaction to complete. When the timer expires, the MIH breaks the connection and declares an error. The connection between the two nodes is marked as operative in error and message IRRC022I is issued.

General errors

This general section contains information about:

- APPC or TCP/IP and the RACF RESTART command
- Deleted nodes

APPC or TCP/IP and the RACF RESTART command

The RESTART command provides a way to recycle or recover a hung or stopped task. See *z/OS Security Server RACF Command Language Reference* for the syntax of the RESTART command.

The preferred way to recycle a connection is not to use the RESTART command. Instead, the operator should use the TARGET command to make the connection to the node dormant and then operative. This method fails if the task is hung waiting for outstanding work. Use the RESTART command to recycle a connection only after an unsuccessful attempt at using the TARGET command to make that connection dormant and then operative.

On restart, the partner node always marks the sender as dormant-remote. The node being restarted then attempts to handshake with the partner to restore the connection.

Deleted node

A command can be executing in the RACF subsystem address space and the node definition for the node on which the command originated can be removed. While requiring the connection to the node to be dormant before it can be deleted reduces the probability of this occurring, it is still possible.

If a message is targeted to a node that is no longer defined to RRSF, the following actions occur:

- A symptom record is created.
- The command output or the command is discarded.

Recording of errors

Environmental errors and transmission errors must be reported to allow for their correction. When the RACF subsystem address space is unable to access a requested system resource, it records this type of error. Further, the RACF subsystem address space attempts to notify the originator of the command that it was unsuccessful. It might also need to stop transmitting to the affected node. The following mechanisms are used:

- A symptom record is created on the system experiencing the failure and stored in the LOGREC data set.
- If possible, an error message reporting the failure is sent back in the RRSFLIST data set (standard returned output) when the failure is not in RRSF. If the failure is in RRSF, then only a symptom record is created. This returned output is an error message. If the failing request is in storage, the failing request is added to the output. To the system where the request originated, the returned output is identical to the output returned when a command has successfully processed.
- RRSF tries the error again after creating the appropriate record of the failure.

Creation of LOGREC data set entries

When the RACF subsystem address space requests an MVS system resource or uses an MVS system service and is unsuccessful, this failure is recorded. If the failure is not recorded by MVS, RACF puts the symptom records in the LOGREC data set.

z/OS MVS Programming: Assembler Services Guide categorizes information contained within the symptom records as:

- Component Data – This identifies the application in which the error occurred (that is, RACF subsystem address space).
- Primary SDB symptoms – This contains the primary string of problem symptoms. This data is used for duplicate problem recognition (See Table 32 on page 148).
- Secondary SDB symptoms – This contains any additional diagnostic values saved at the time of the error.
- Variable Data – This contains diagnostic data, such as portions of data areas or parameter lists pertinent to the error.

Table 32 on page 148 shows the contents of the primary symptom string that the symptom record contains.

Table 32. Contents of primary symptom string for a RACF subsystem address space error

SDB key literal	Usage
RIDS/	Module CSECT name
RIDS/	Load module name
PCSS/	System service (MVS or RACF macro) that failed
PRCS/	Return code from service
PRCS/	Reason code from service (if no reason code is available, this field contains a value of zero)

Table 33 on page 149 shows the contents of the secondary symptom string that the symptom record contains.

Table 33. Contents of secondary symptom string for a RACF subsystem address space error

SDB key literal	Usage
ADRS/	Offset of service that failed

Table 34 on page 149 shows the contents of the variable symptom string that the symptom record contains.

Table 34. Potential contents of variable symptom string for a RACF subsystem address space error for the indicated protocol used

Field	Description
LUNAME	APPC partner LUNAME
TPNAME	APPC partner TPNAME
MODENAME	APPC partner MODE NAME
ADDRESS	TCP partner host address
PORTNUM	TCP listener port number
MISC STRING	Additional information for some service failures

Table 35 on page 149 shows the additional information that the symptom record contains.

Table 35. Additional LOGREC data for a RACF subsystem address space error

Field	Value
COMPDSC	RACF subsystem address space
PROBLEM	Name of failing service
SERVLEV	Service level of module
COMPID	Component ID (5752XXH00)
COMPLEV	Component level (260)

Chapter 6. Diagnosis reference for RACF

This topic contains:

- Parameter list descriptions (see “Parameter list descriptions” on page 151)
- Process flows (see “Process flows for RACF” on page 182)

Parameter list descriptions

This describes parameter lists passed to RACF when a RACF macro is invoked. Installation exits, which are described in *z/OS Security Server RACF System Programmer's Guide*, also use parameter lists, which are described in *z/OS Security Server RACF Data Areas* in the z/OS Internet library (www.ibm.com/servers/resourcelink/svc00100.nsf/pages/zosInternetLibrary). See Appendix A, “The Inventory Control Block (ICB),” on page 243 for information about the ICB (inventory control block).

The parameter lists used by RACF in processing RACF commands and the RACF report writer are standard TSO/E parameter lists. See *z/OS TSO/E Customization* for information about these parameter lists.

The parameter lists used by RACF in processing RACF utilities and DSMON are standard MVS parameter lists.

Table 36 on page 151 is a quick reference that identifies the system macro-request types that are replacements for the independent RACF system macros. For more information about both types, see *z/OS Security Server RACROUTE Macro Reference*.

Table 36. Cross-reference for RACROUTE REQUEST=type, the independent RACF system macros, and the parameter list name

RACROUTE REQUEST=*type*	**Independent RACF system macro**	**Parameter list**
REQUEST=AUDIT	None	AUDIT
REQUEST=AUTH	RACHECK	AUTH
REQUEST=DEFINE	RACDEF	DEFINE
REQUEST=DIRAUTH	None	DIRAUTH
REQUEST=EXTRACT	RACXTRT	EXTRACT
REQUEST=FASTAUTH	FRACHECK	FASTAUTH
REQUEST=LIST	RACLIST	LIST
REQUEST=SIGNON	None	SIGNON
REQUEST=STAT	RACSTAT	STAT
REQUEST=TOKENBLD	None	TOKENBLD
REQUEST=TOKENMAP	None	TOKENMAP
REQUEST=TOKENXTR	None	TOKENXTR
REQUEST=VERIFY	RACINIT	VERIFY
REQUEST=VERIFYX	None	VERIFYX

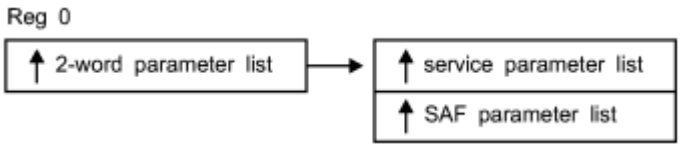
The following are the parameter lists:

- “RACROUTE parameter list” on page 152
- “AUDIT service parameter list” on page 155
- “AUTH service parameter list” on page 156
- “DEFINE service parameter list” on page 160
- “DIRAUTH service parameter list” on page 165
- “EXTRACT service parameter list” on page 166

- “FASTAUTH service parameter list” on page 168
- “LIST service parameter list” on page 169
- “SIGNON service parameter list” on page 170
- “STAT service parameter list” on page 171
- “TOKENBLD service parameter list” on page 172
- “TOKENMAP/TOKENXTR service parameter list” on page 175
- “VERIFY/VERIFYX service parameter list” on page 176
- “ICHEINTY parameter list” on page 180

“Checking the parameter list passed to RACF” on page 81 goes into detail to show you how to locate parameters in a dump. To do this, you find the contents of register 1.

If register 1 is zero, then the RACROUTE macro was issued, and module ICHRFR00 issued the RACF macro. In this case, register zero points to a two-word parameter list. The first word of this parameter list points to the RACF SVC parameter list, and the second word of this parameter list points to the SAF parameter list. This is shown in the following:



If register 1 is not zero, then this is the address of the parameter list passed with the macro request.

RACROUTE parameter list

The RACROUTE parameter list is also called the SAF router parameter list or the MVS router parameter list. It is mapped by macro ICHSAFP. For another description of this parameter list that includes field names, see *z/OS Security Server RACF Data Areas* in the z/OS Internet library (www.ibm.com/servers/resourcelink/svc00100.nsf/pages/zosInternetLibrary).

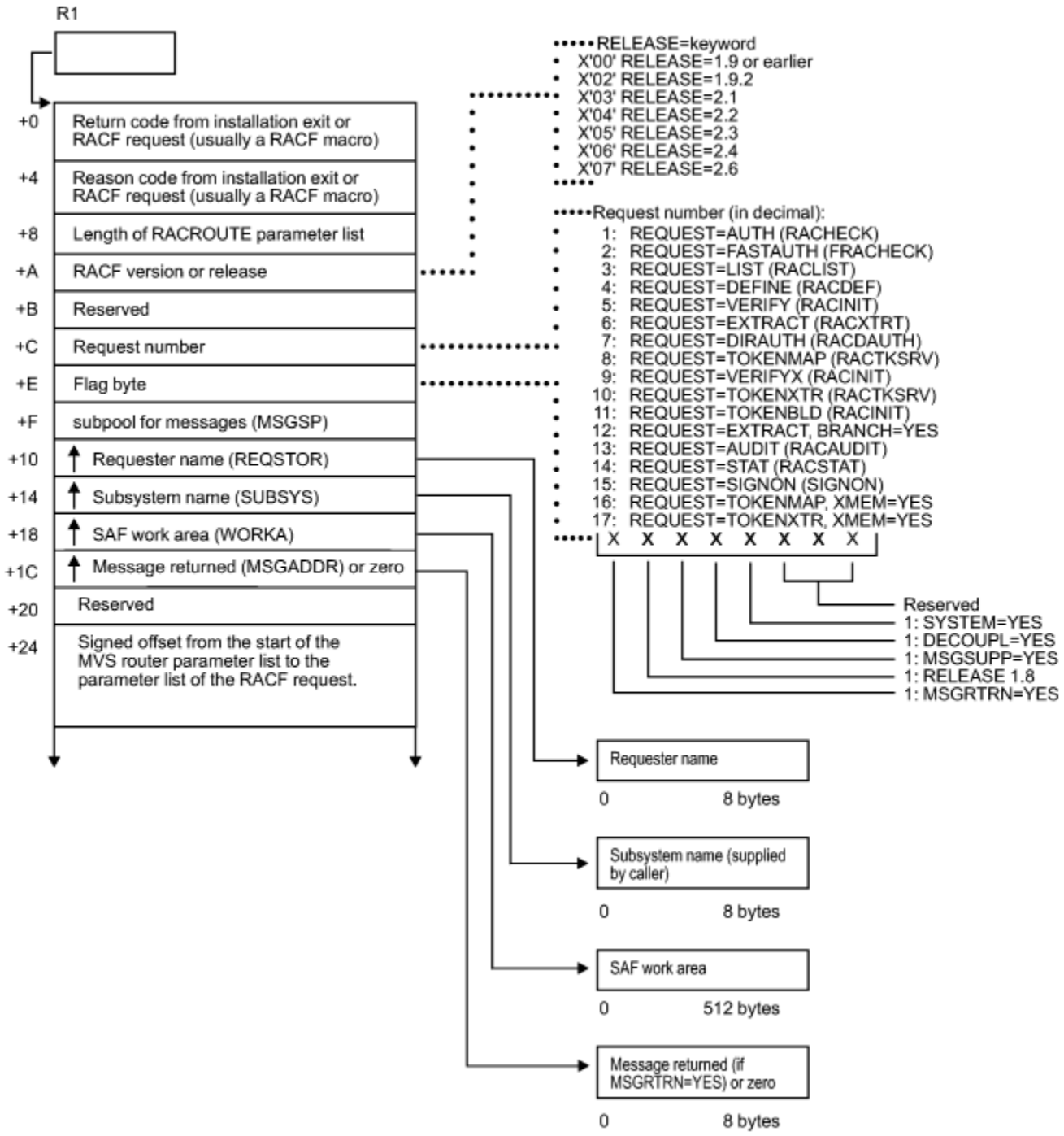


Figure 17. RACROUTE parameter list (part 1 of 2)

These areas show up on both MVS and VM systems, but they are used only on VM.

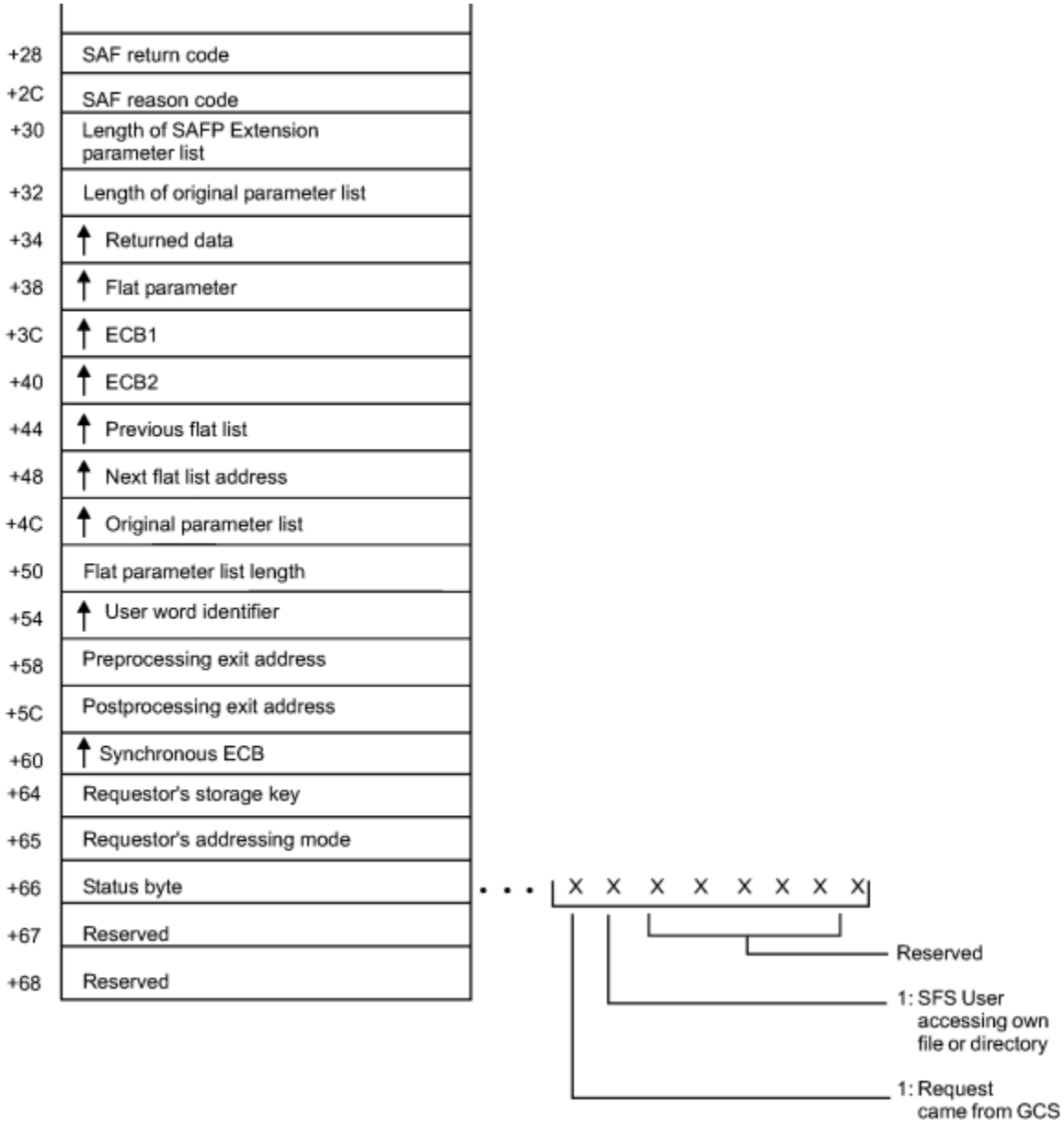


Figure 18. RACROUTE parameter list (part 2 of 2)

AUDIT service parameter list

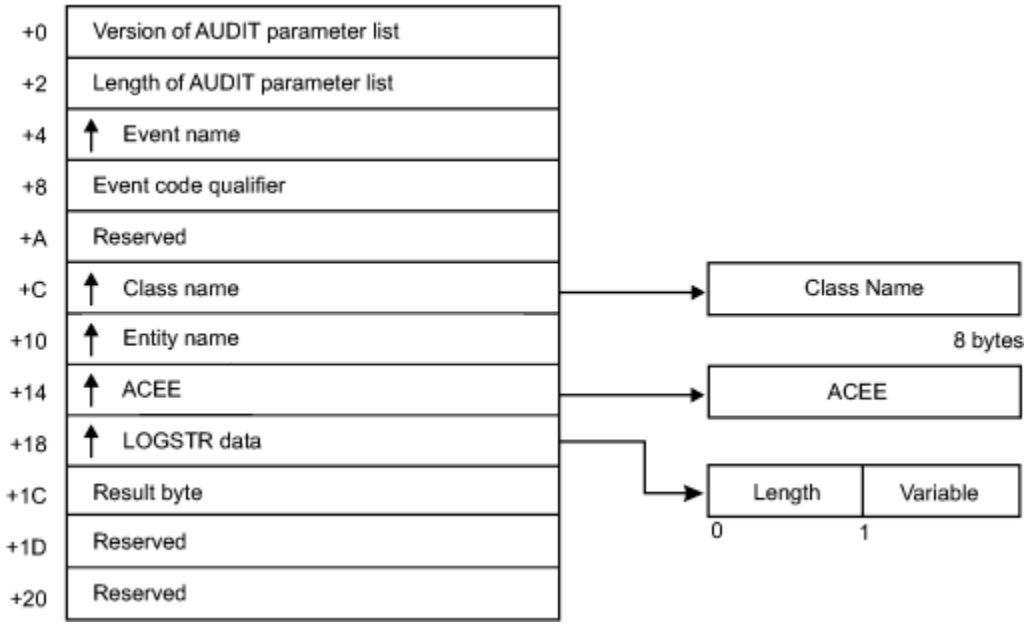
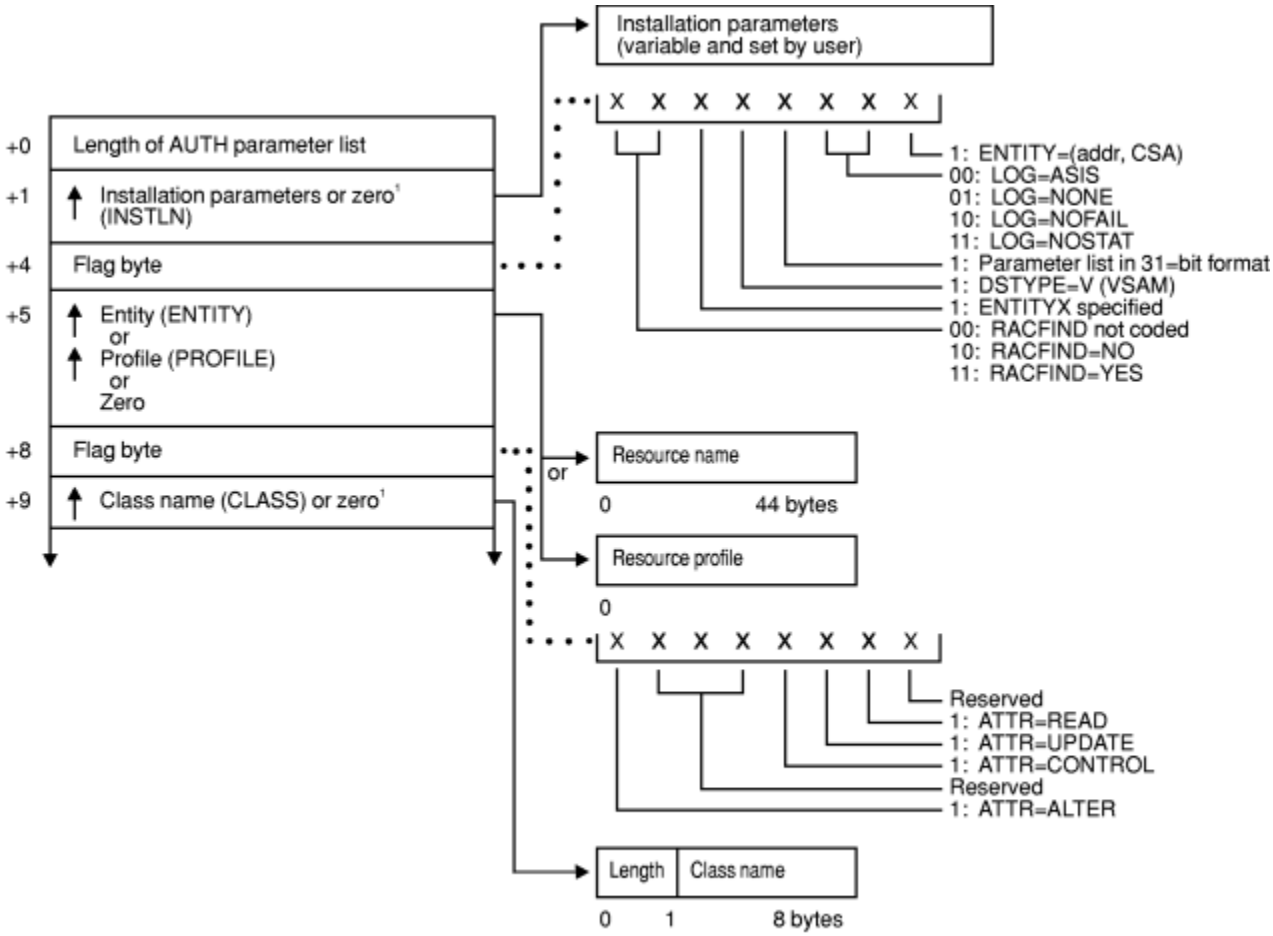


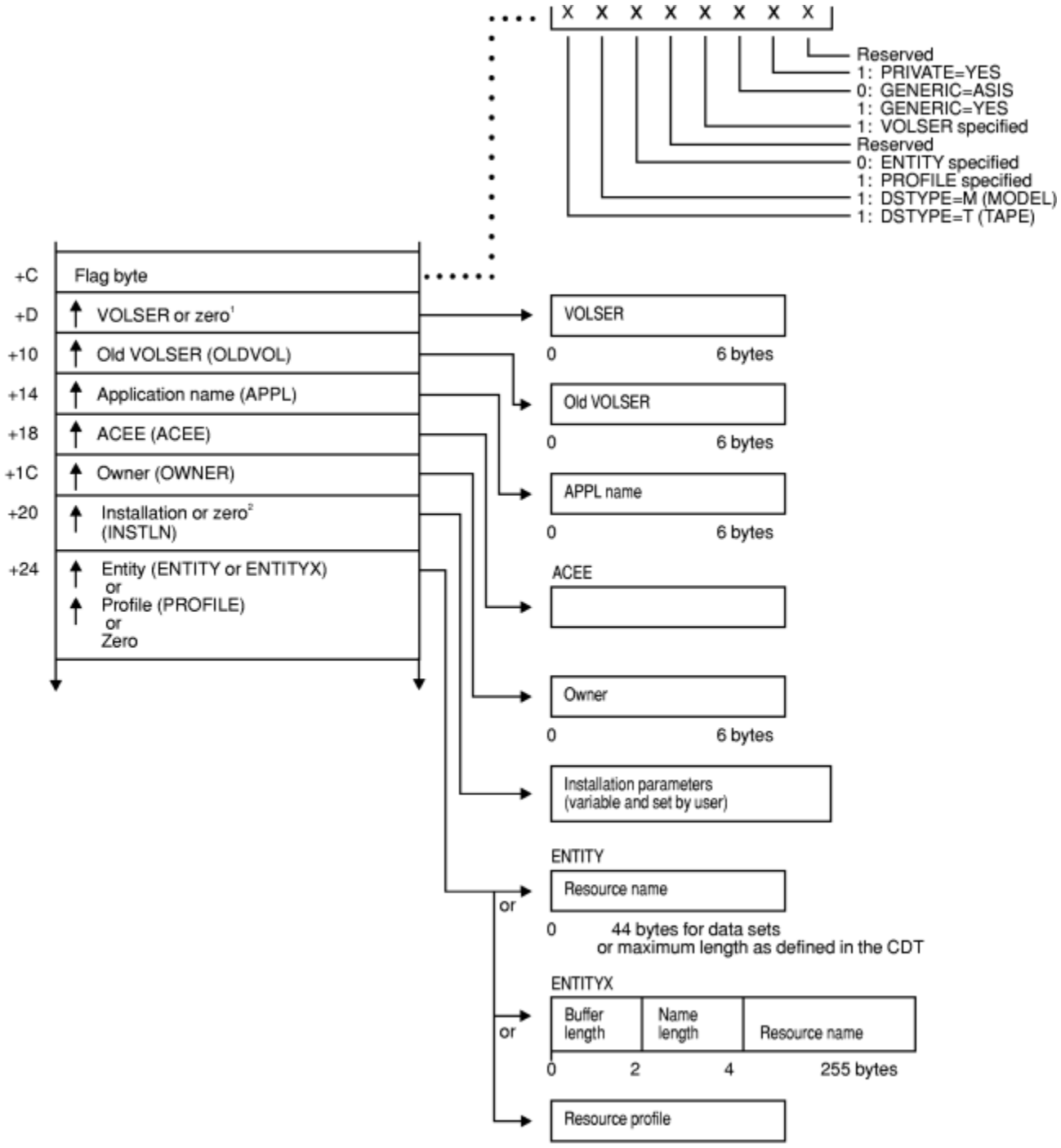
Figure 19. AUDIT service parameter list

AUTH service parameter list



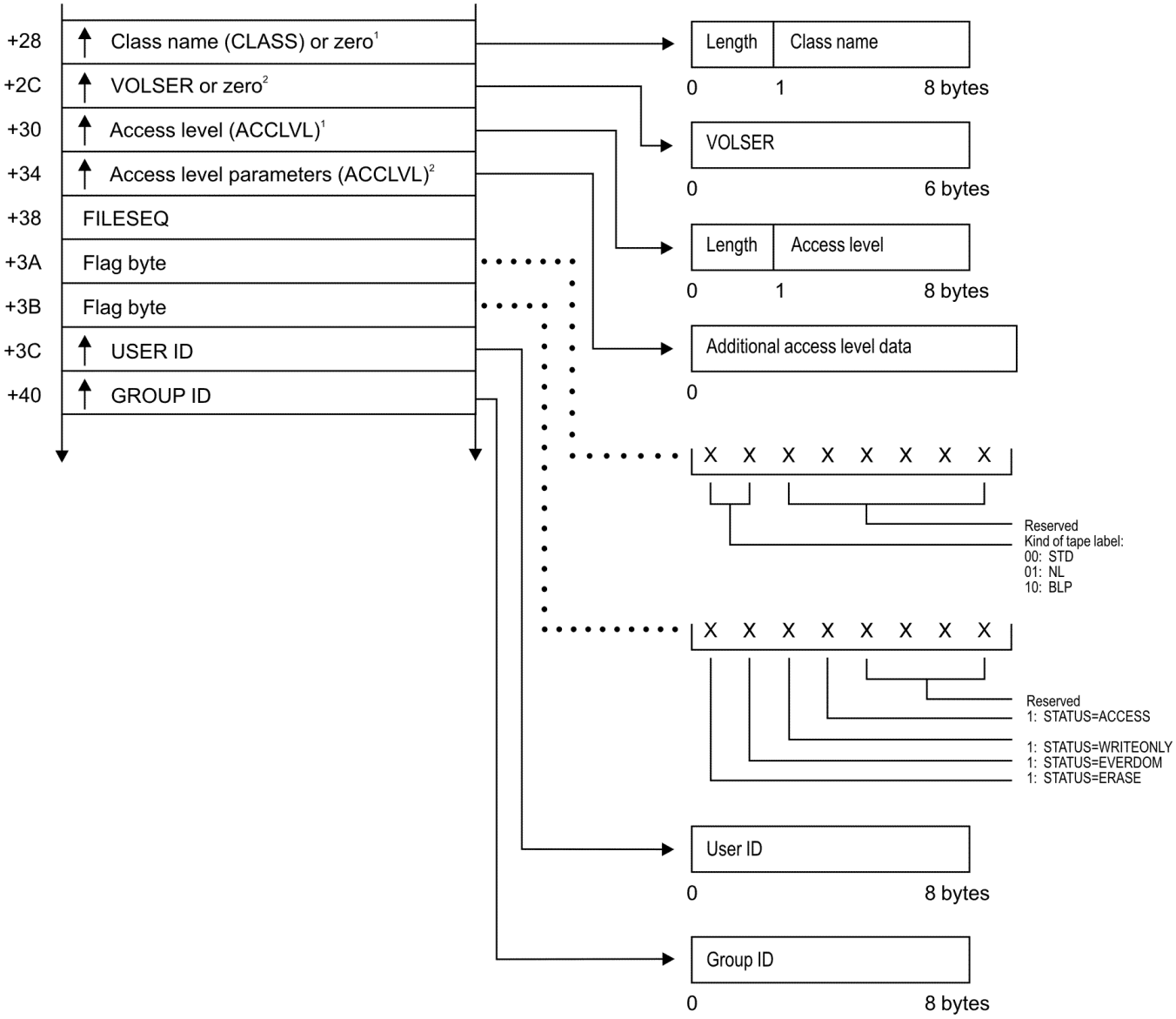
[1] This field is zero if bit 4 of the flag byte at offset 4 is one (input was RACROUTE REQUEST=AUTH).

Figure 20. AUTH service parameter list (part 1 of 4)



[1] This field is zero if bit 4 of the flag byte at offset 4 is one (input was RACROUTE REQUEST=AUTH).
[2] This field is zero if bit 4 of the flag byte at offset 4 is zero (input was RACHECK macro).

Figure 21. AUTH service parameter list (part 2 of 4)



[1] This field is zero if bit 4 of the flag byte at offset 4 is one (input was RACROUTE REQUEST=AUTH).
[2] This field is zero if bit 4 of the flag byte at offset 4 is zero (input was RACHECK macro).

Figure 22. AUTH service parameter list (part 3 of 4)

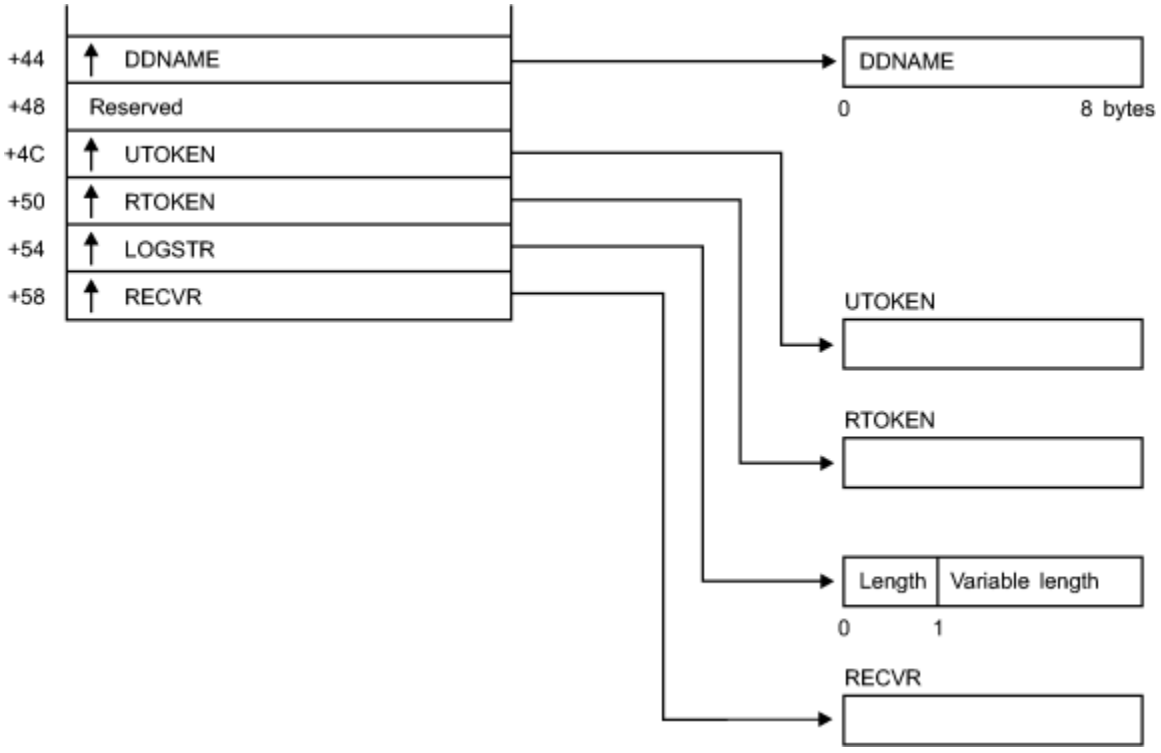
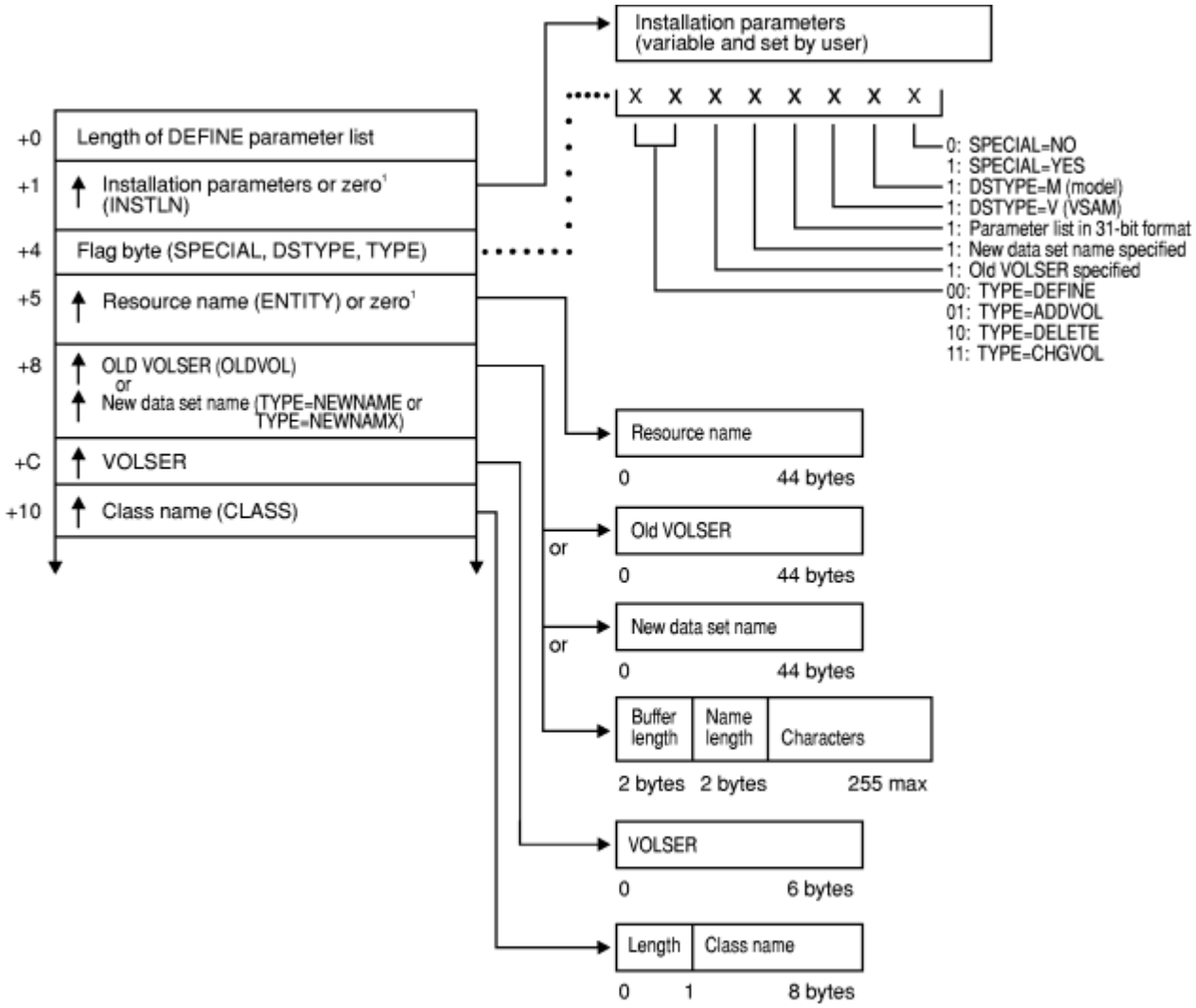


Figure 23. AUTH service parameter list (part 4 of 4)

DEFINE service parameter list



[1] This field is zero if bit 4 of the flag byte at offset 4 is one (input was RACROUTE REQUEST=DEFINE).

Figure 24. DEFINE service parameter list (part 1 of 6)

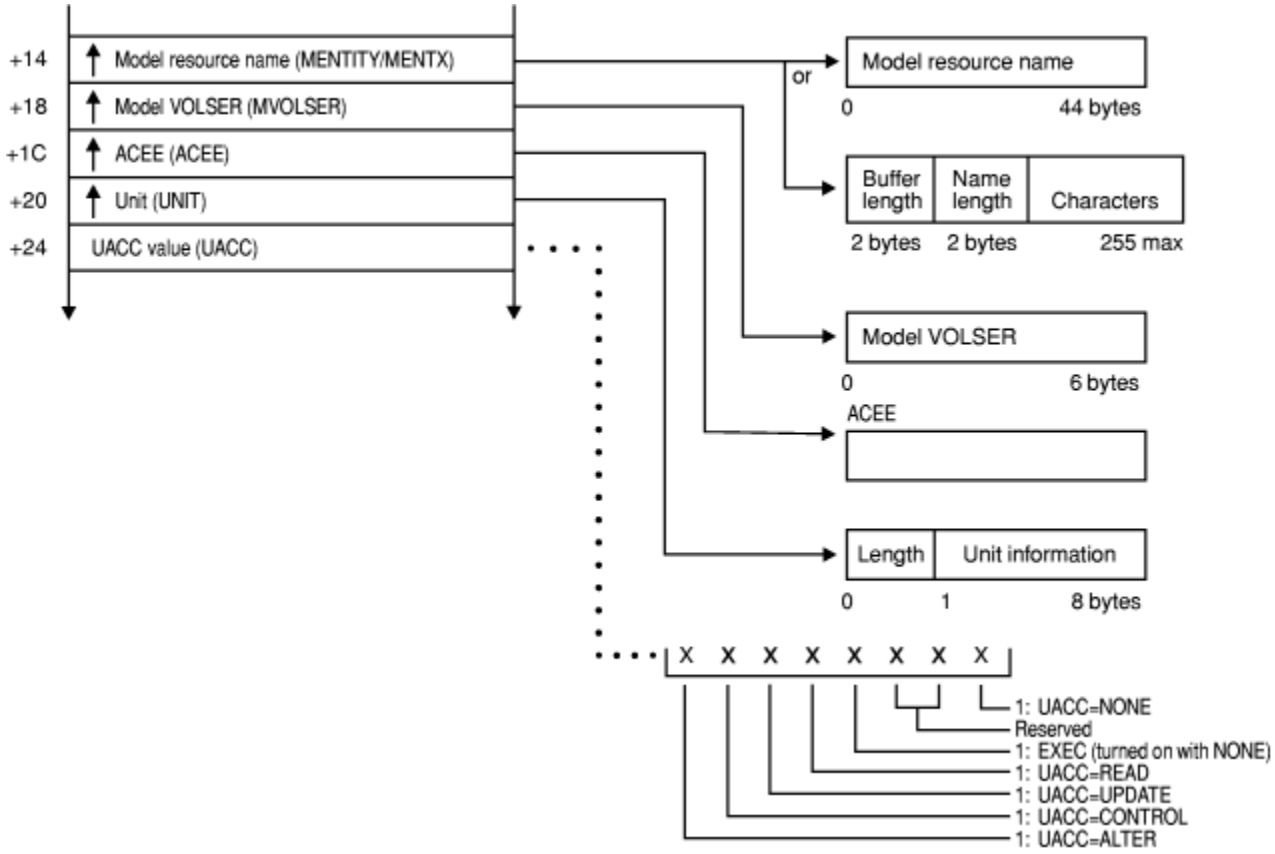


Figure 25. DEFINE service parameter list (part 2 of 6)

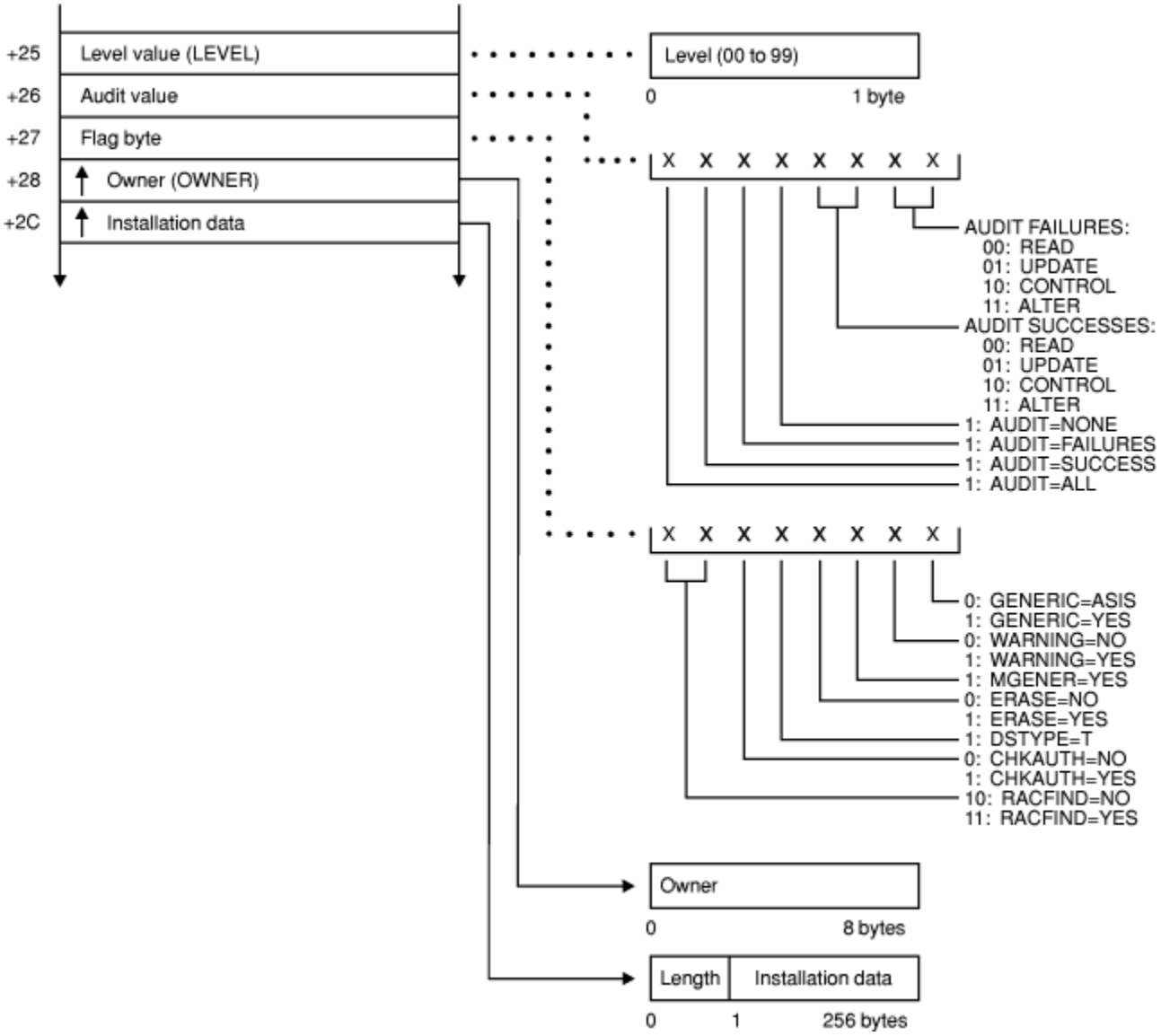
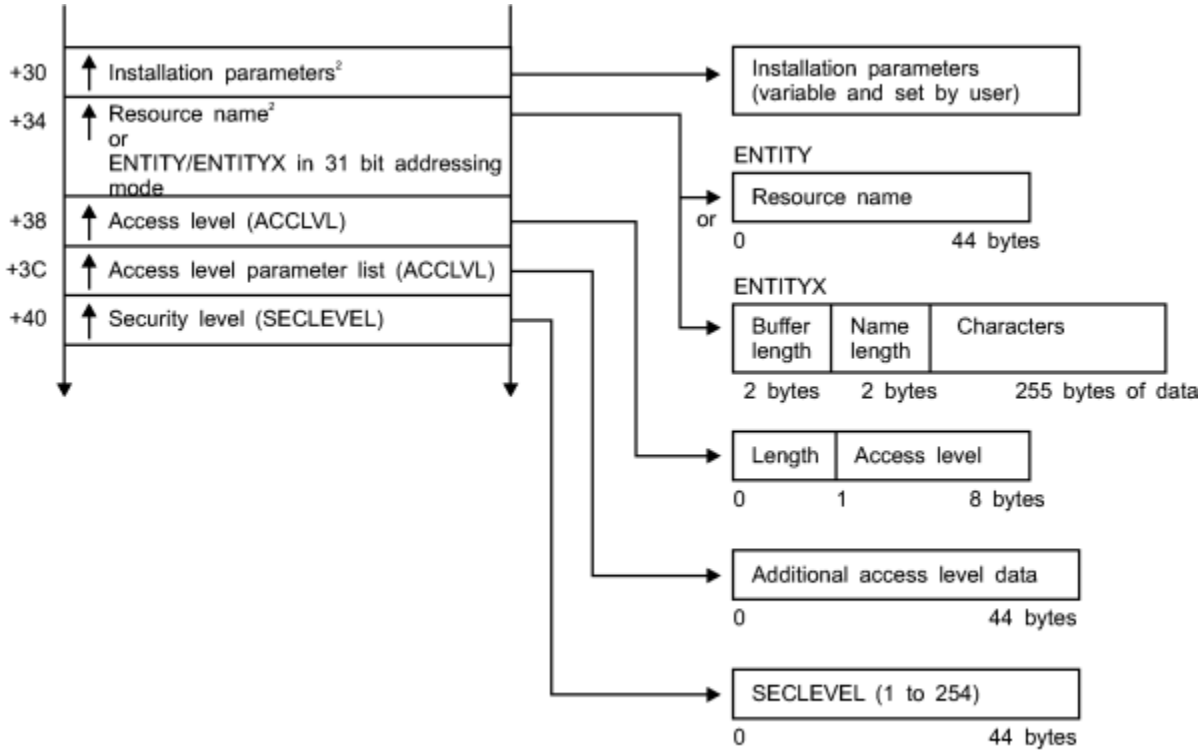


Figure 26. DEFINE service parameter list (part 3 of 6)



[2] This field is zero if bit 4 of the flag byte at offset 4 is zero (input was RACDEF macro).

Figure 27. DEFINE service parameter list (part 4 of 6)

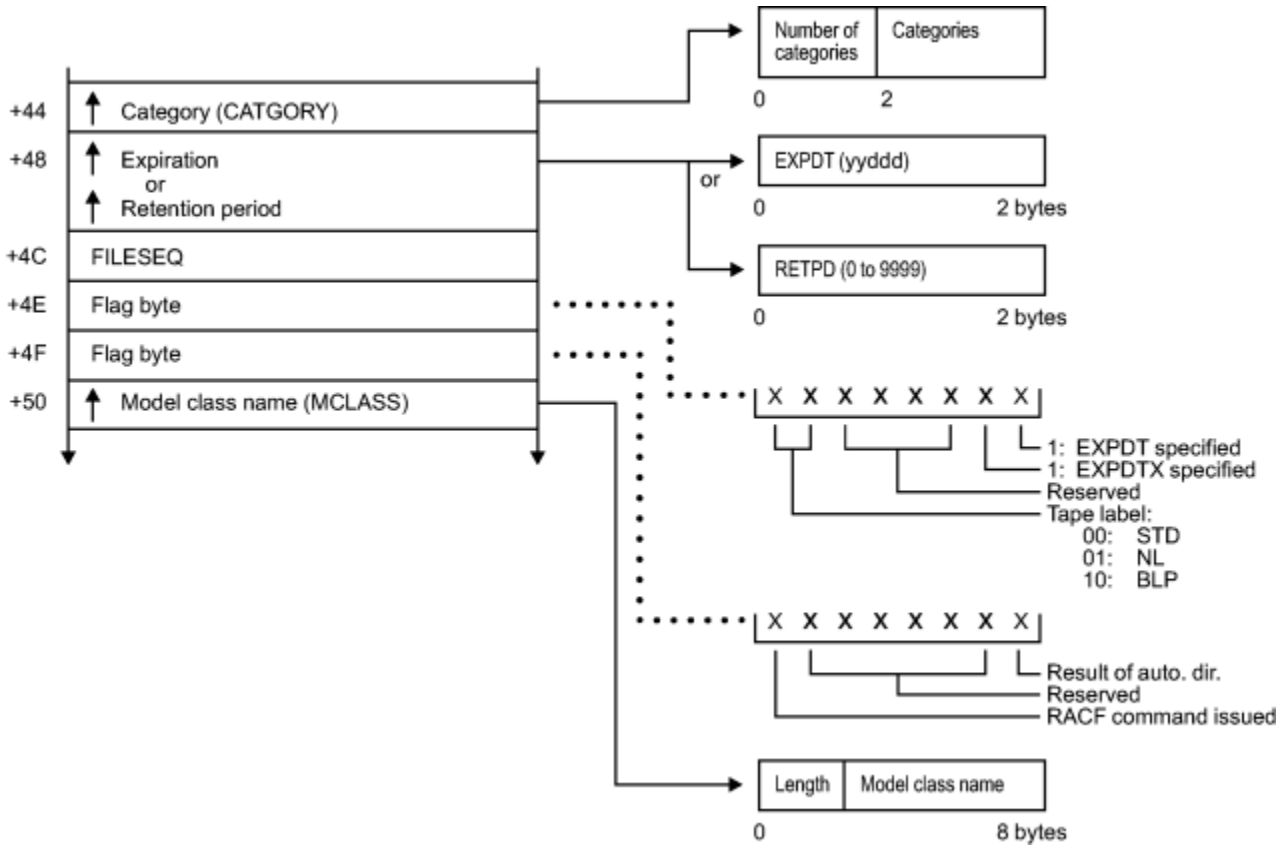


Figure 28. DEFINE service parameter list (part 5 of 6)

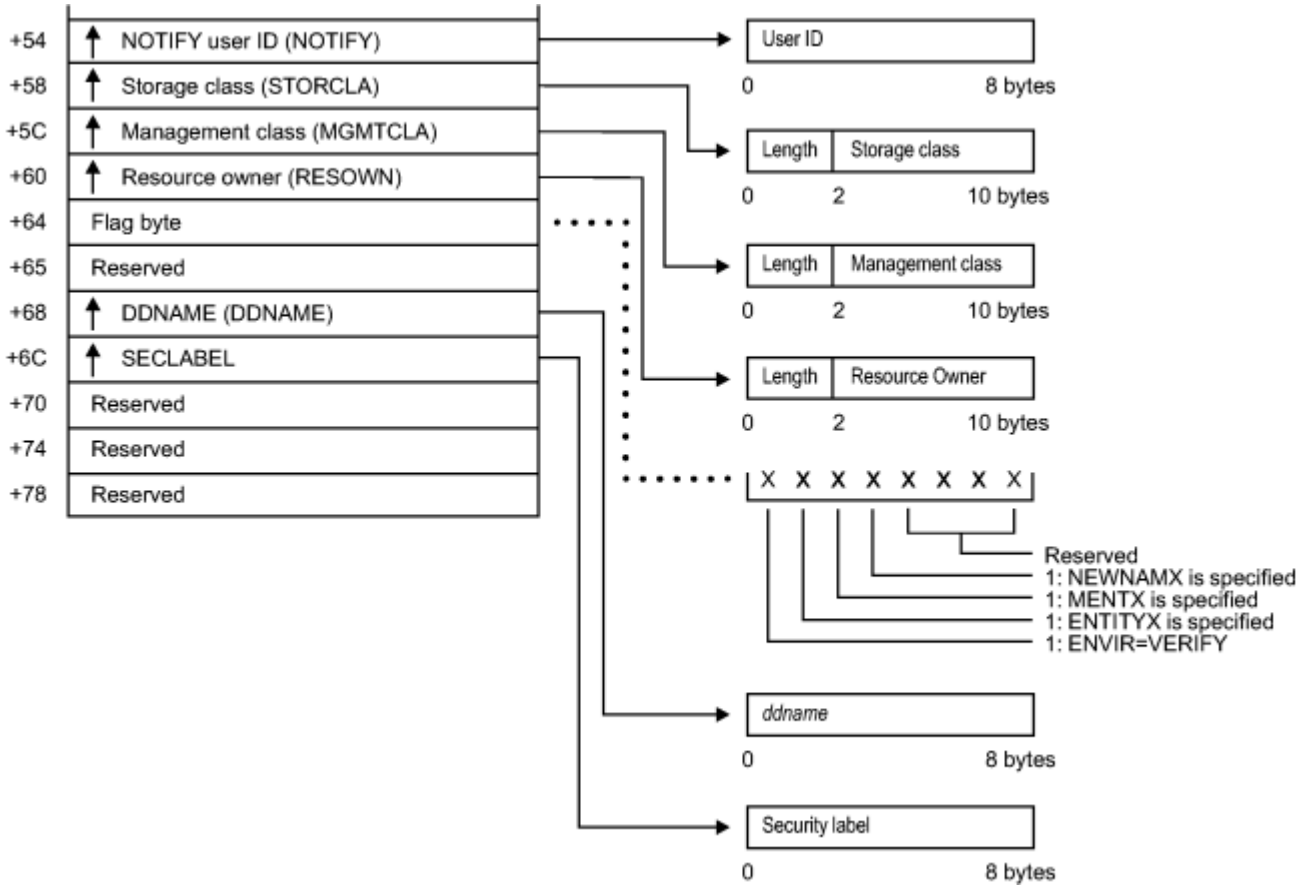


Figure 29. DEFINE service parameter list (part 6 of 6)

DIRAUTH service parameter list

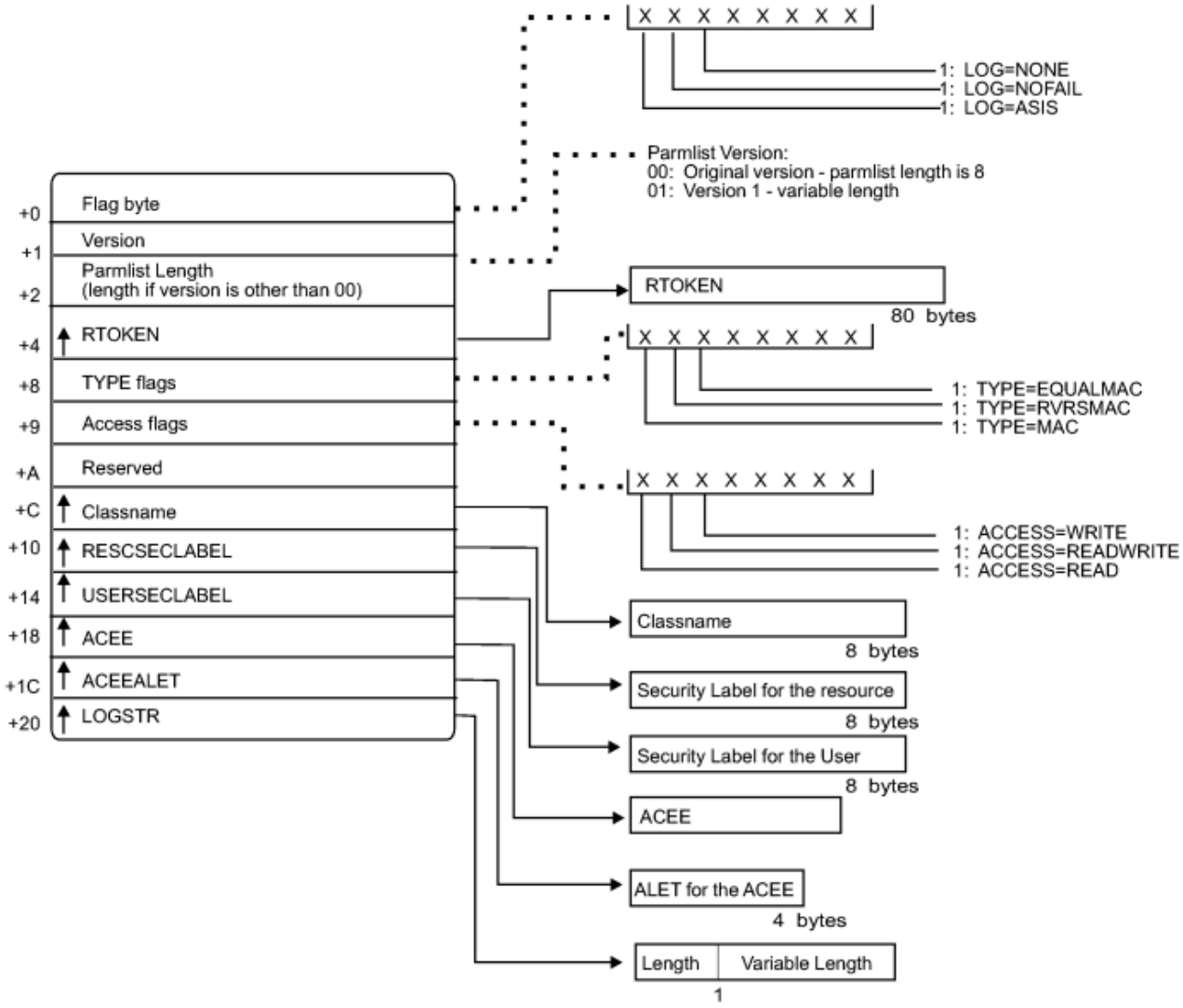


Figure 30. DIRAUTH service parameter list

EXTRACT service parameter list

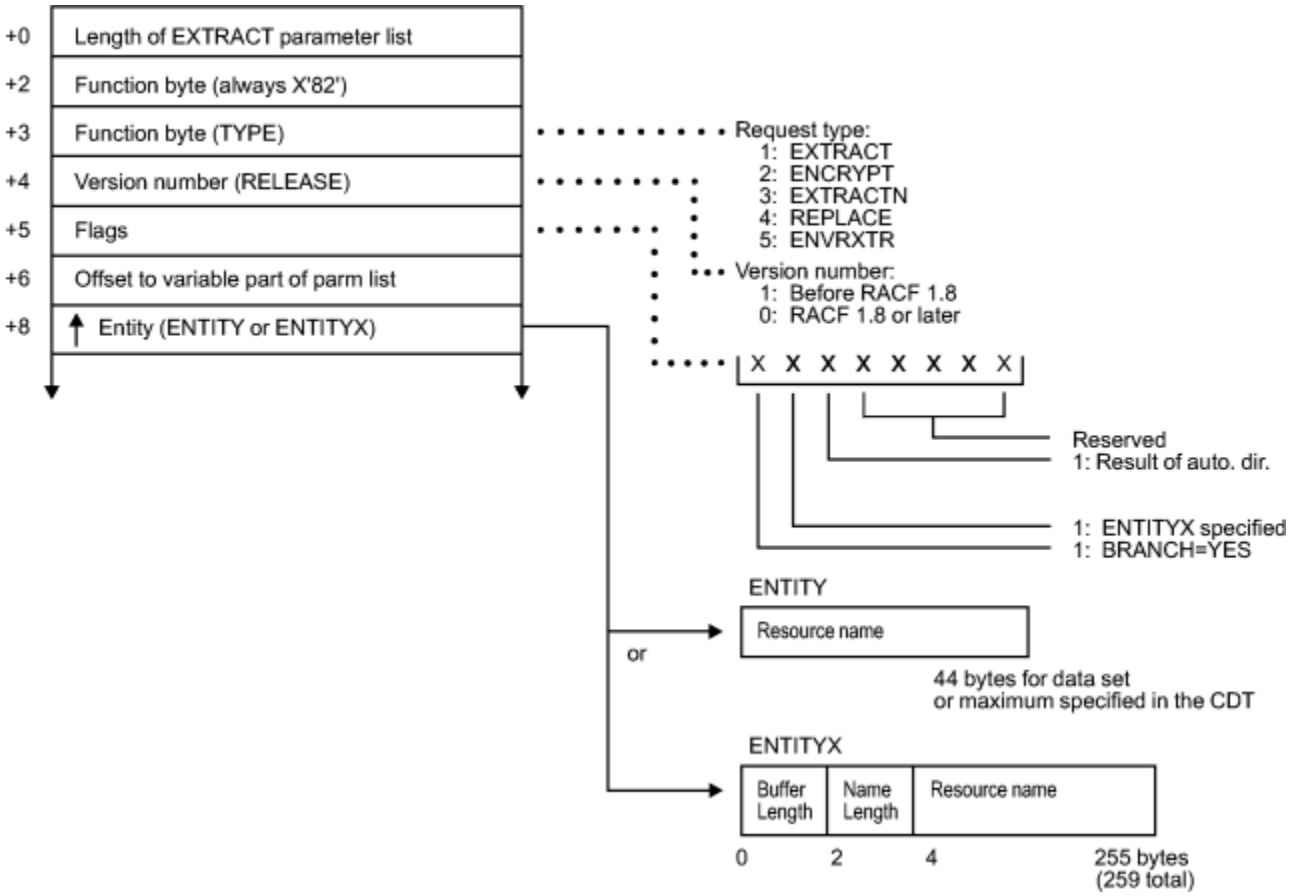


Figure 31. EXTRACT service parameter list (part 1 of 3)

(The following is used when TYPE = ENVRXTR.)

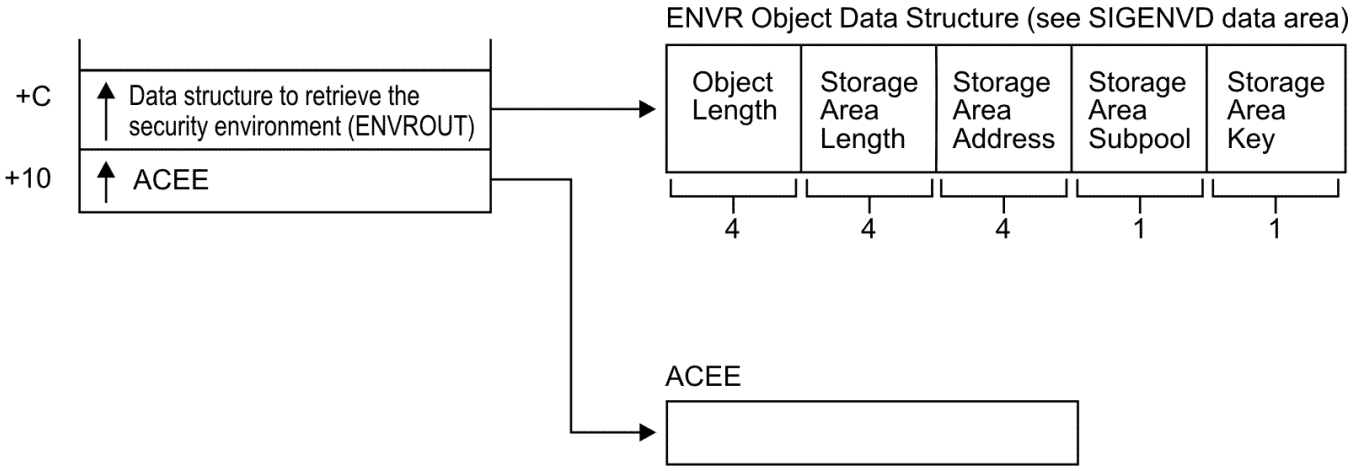


Figure 32. EXTRACT service parameter list (part 2 of 3)

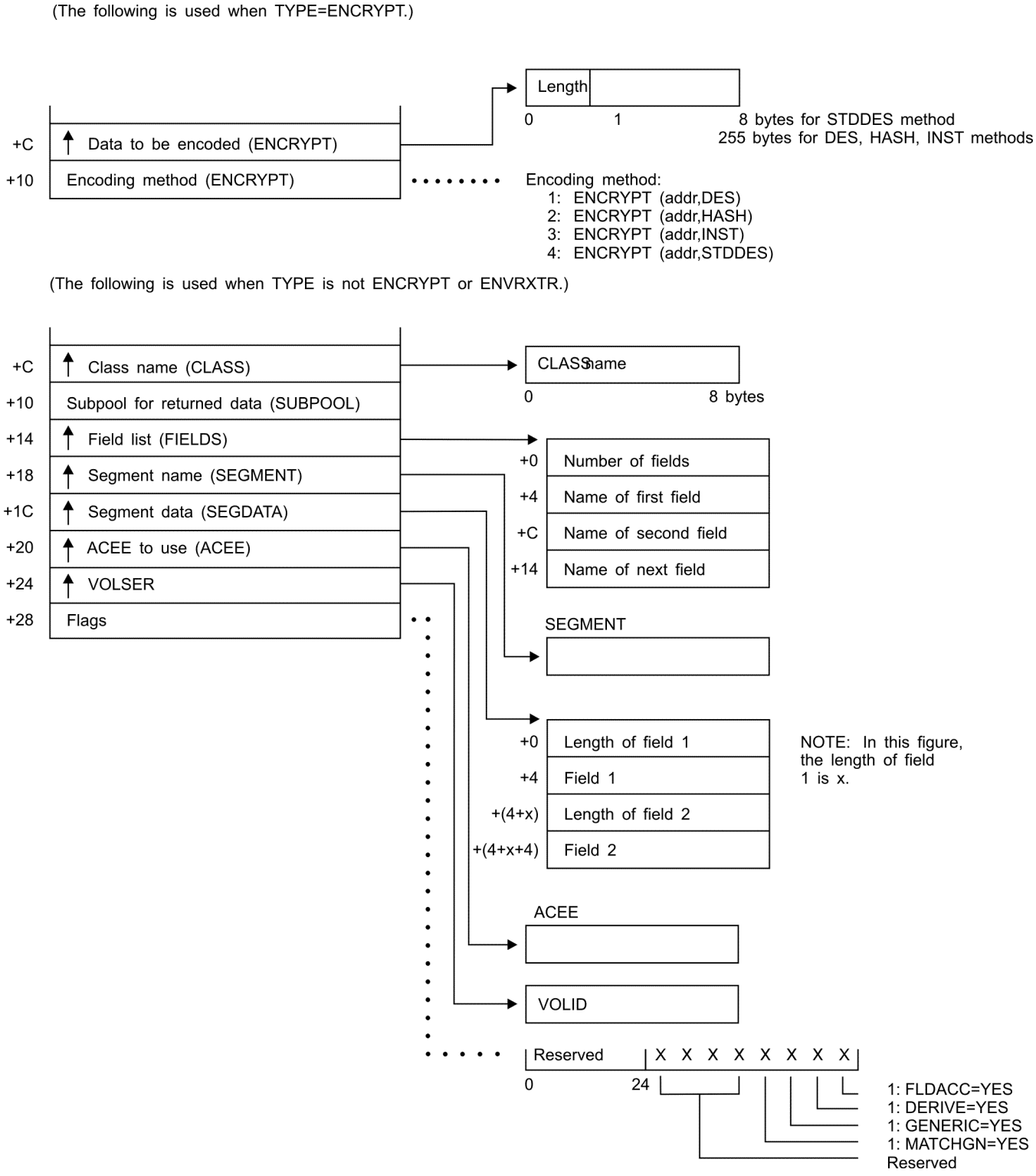


Figure 33. EXTRACT service parameter list (part 3 of 3)

FASTAUTH service parameter list

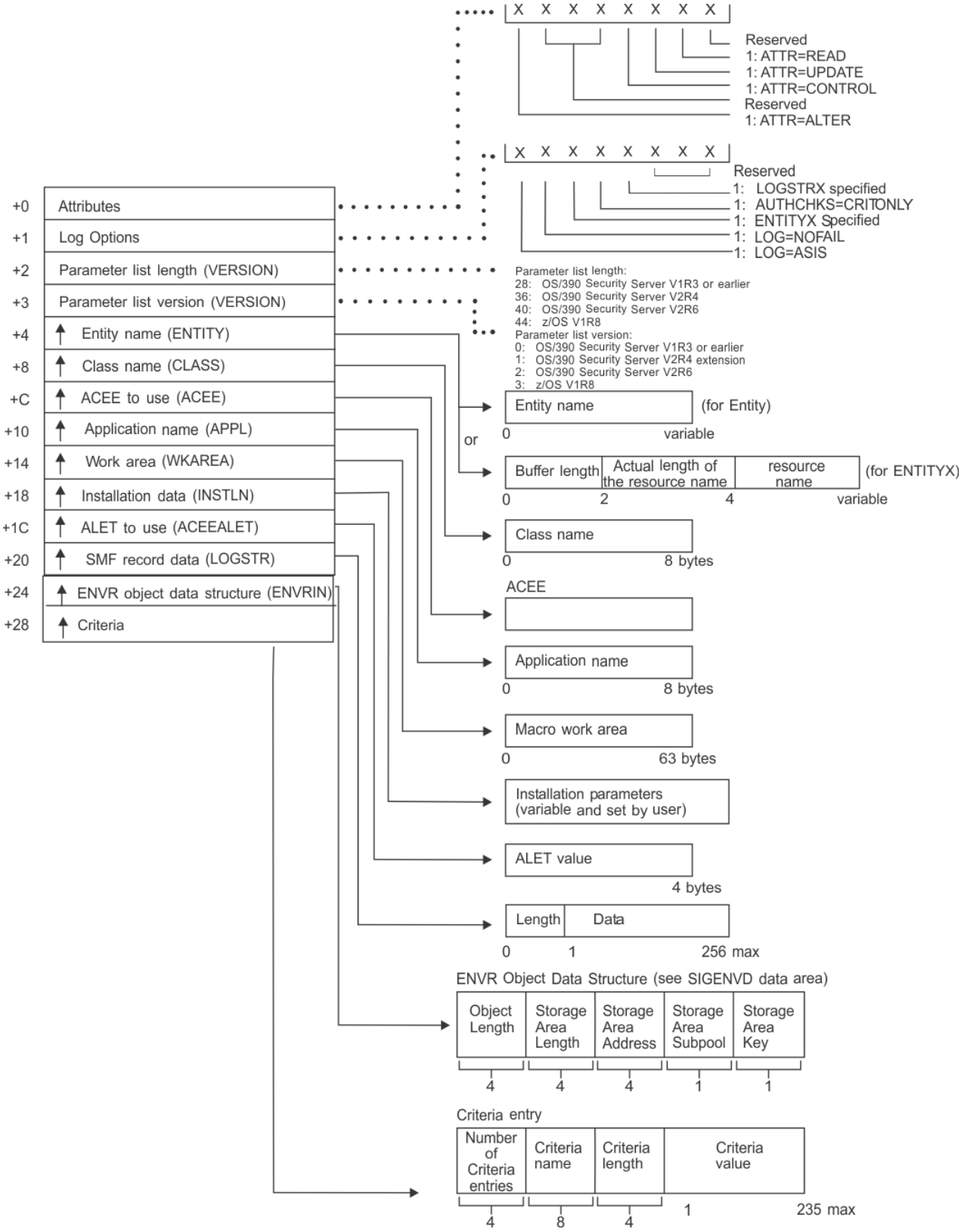


Figure 34. FASTAUTH service parameter list

LIST service parameter list

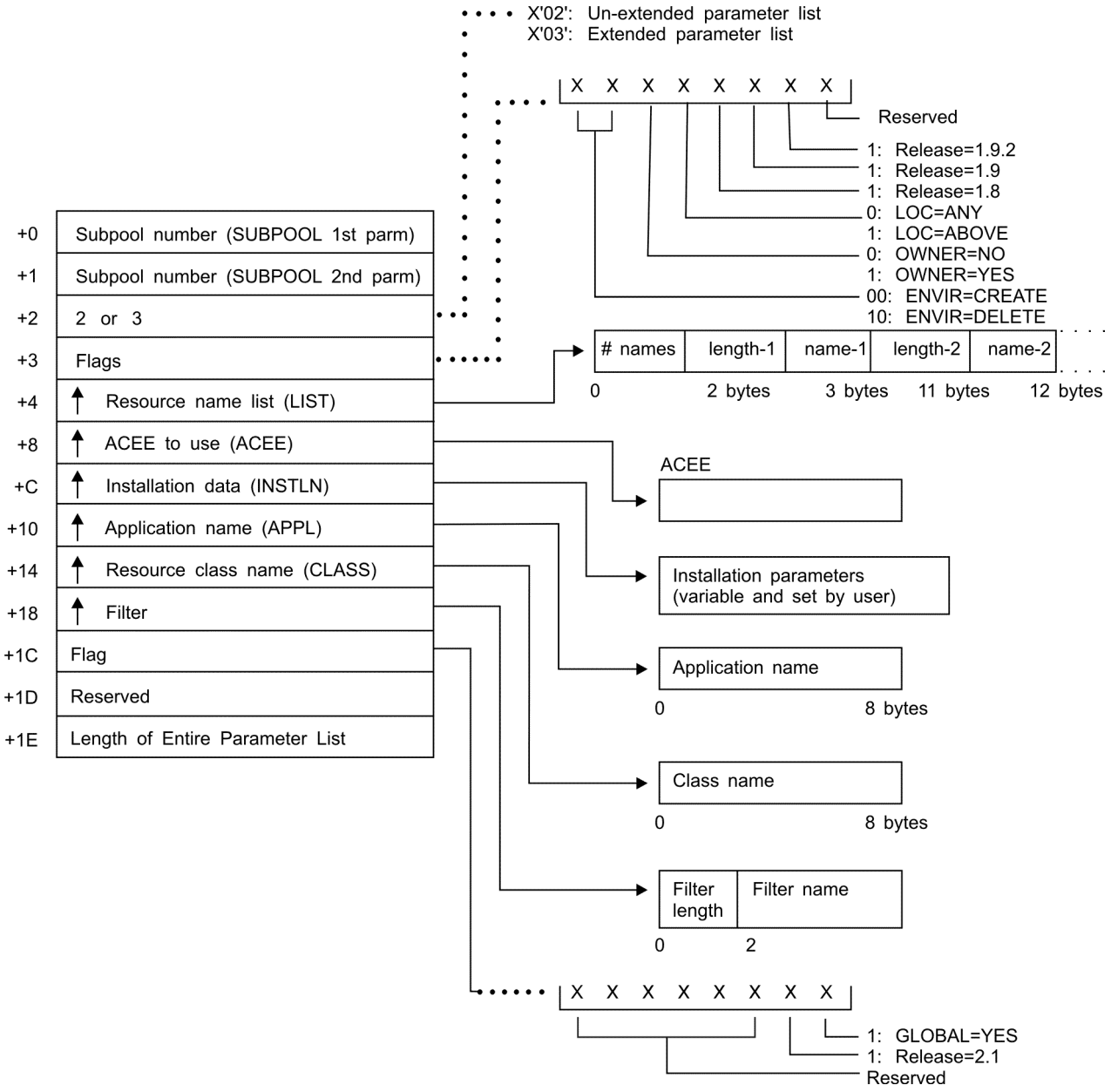


Figure 35. LIST service parameter list

SIGNON service parameter list

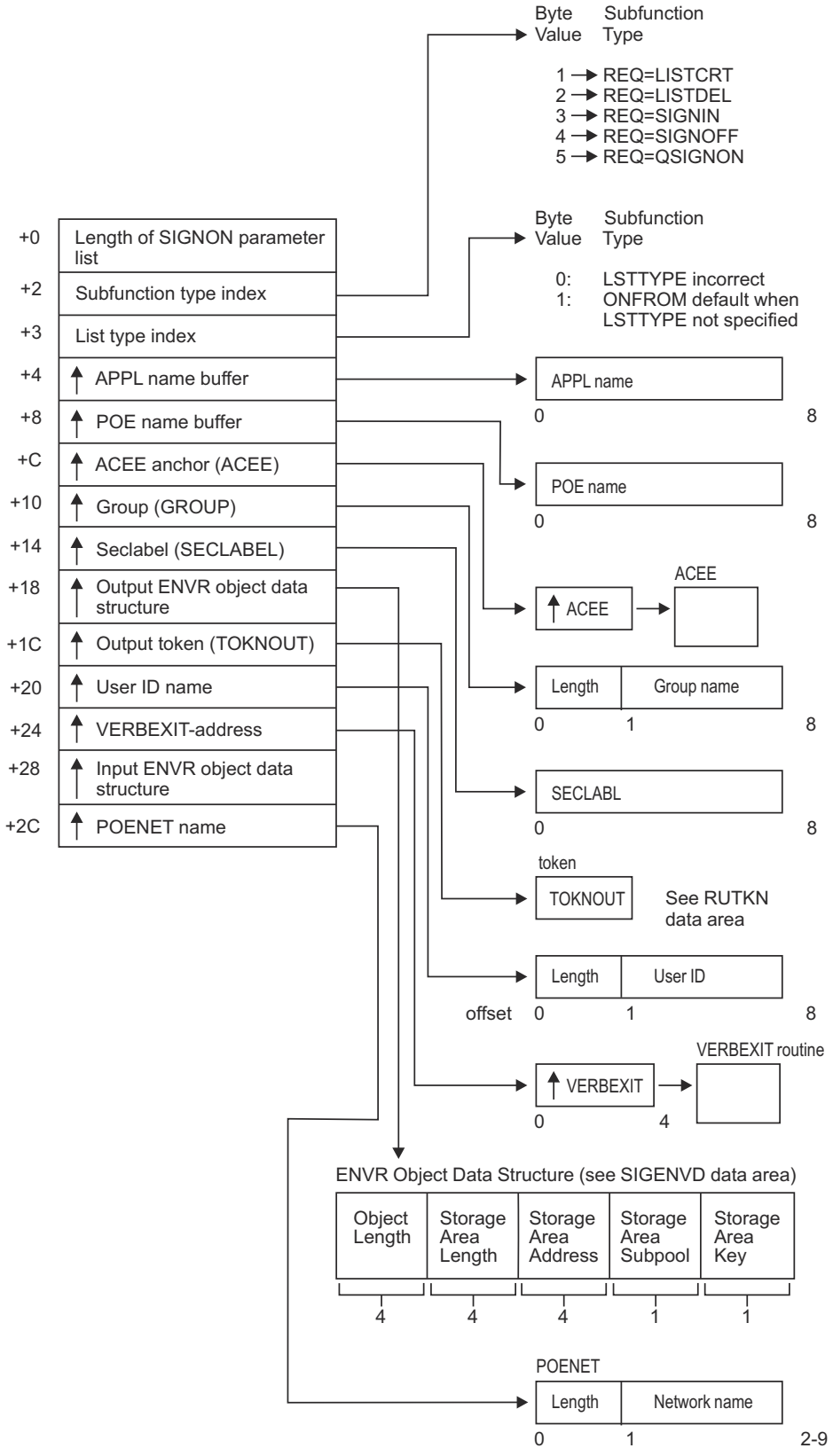


Figure 36. SIGNON service parameter list

STAT service parameter list

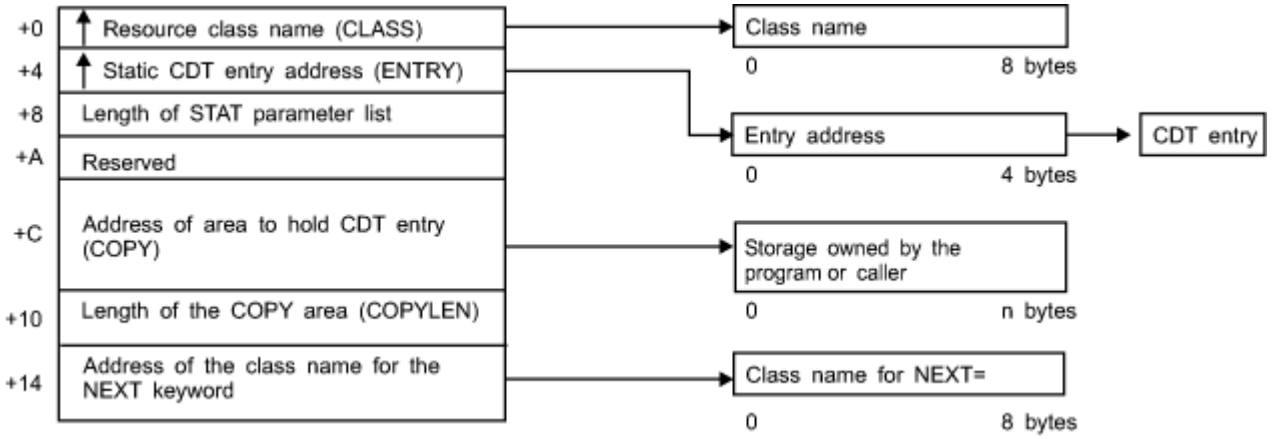


Figure 37. STAT service parameter list

TOKENBLD service parameter list

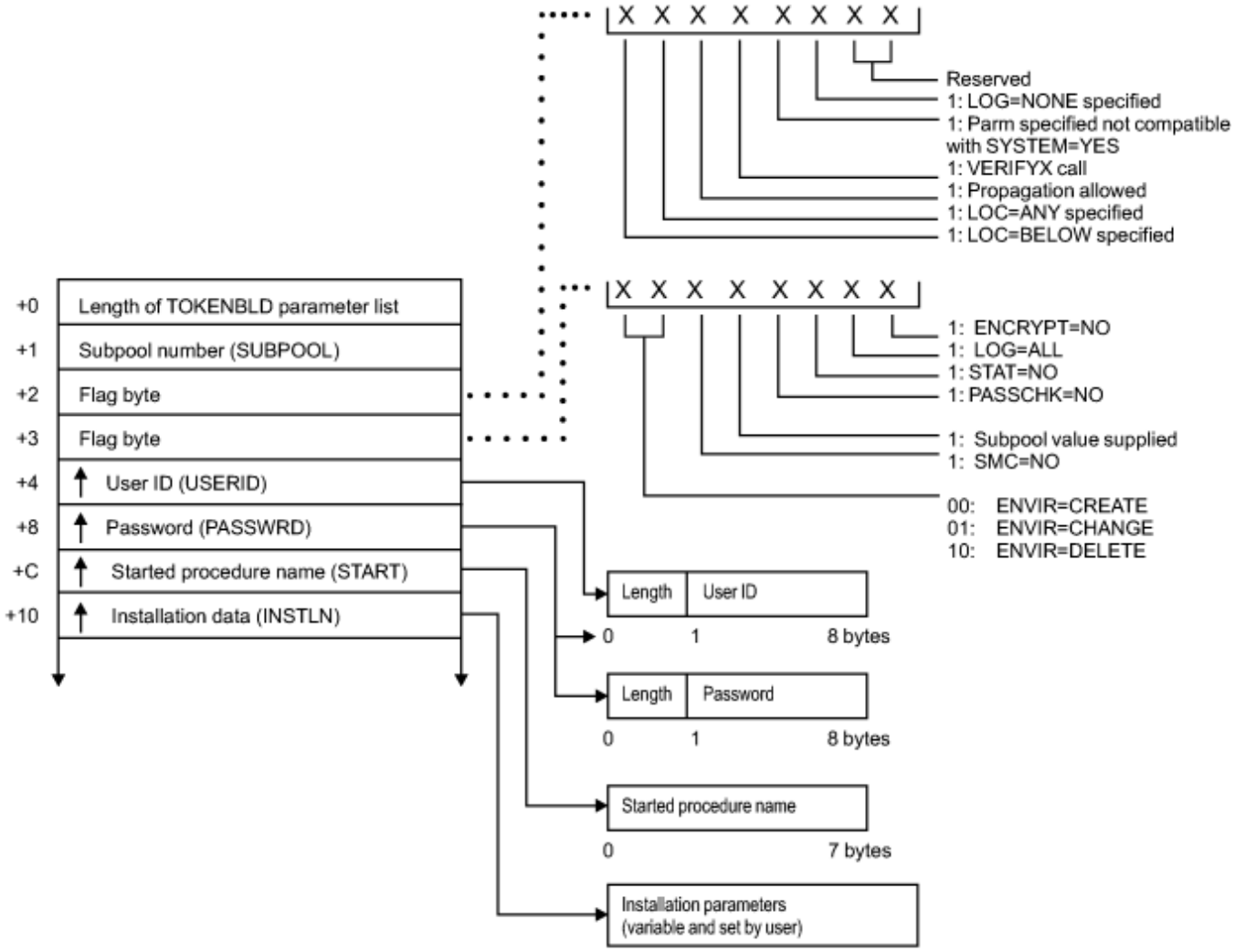


Figure 38. TOKENBLD service parameter list (part 1 of 4)

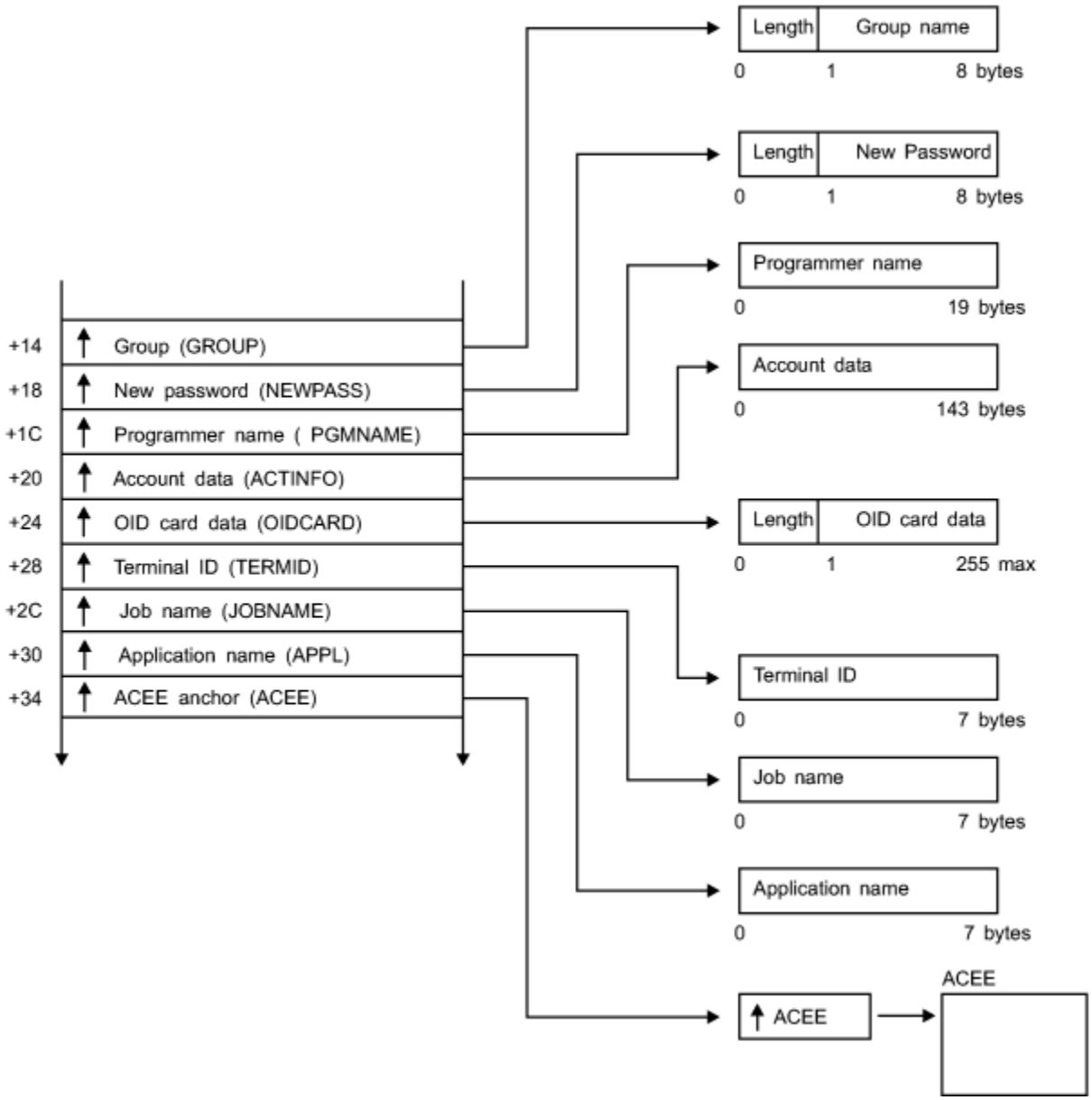


Figure 39. TOKENBLD service parameter list (part 2 of 4)

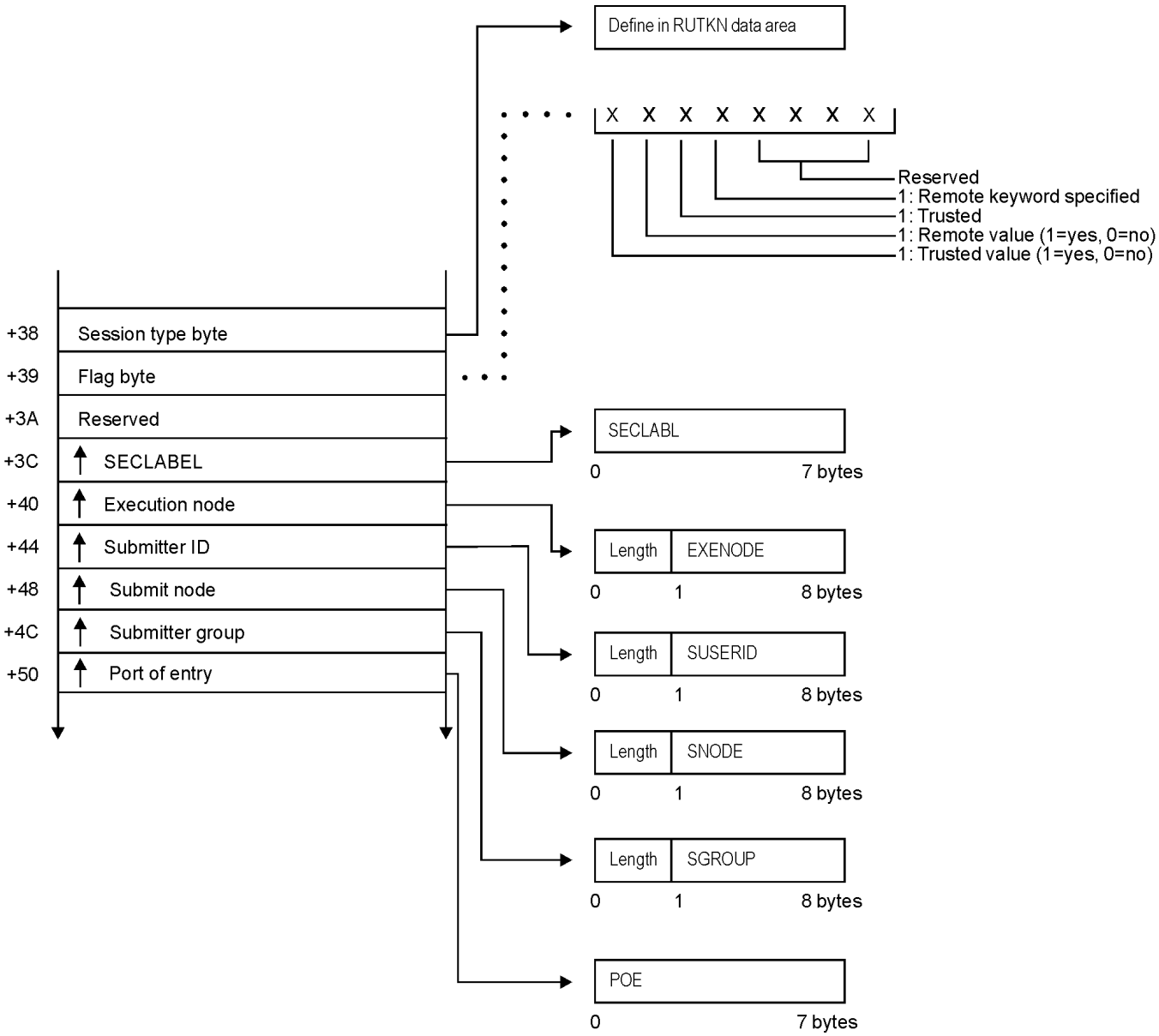


Figure 40. TOKENBLD service parameter list (part 3 of 4)

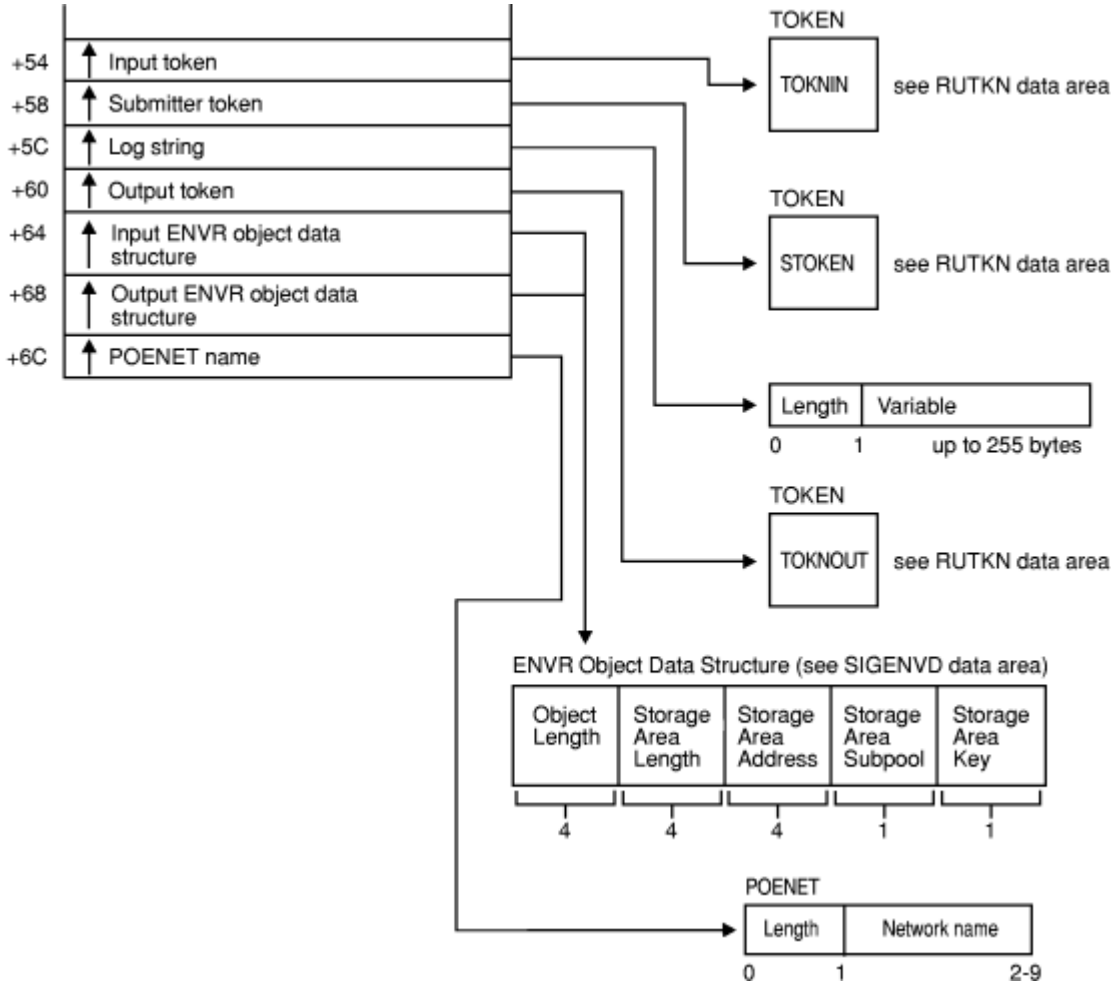


Figure 41. TOKENBLD service parameter list (part 4 of 4)

TOKENMAP/TOKENXTR service parameter list

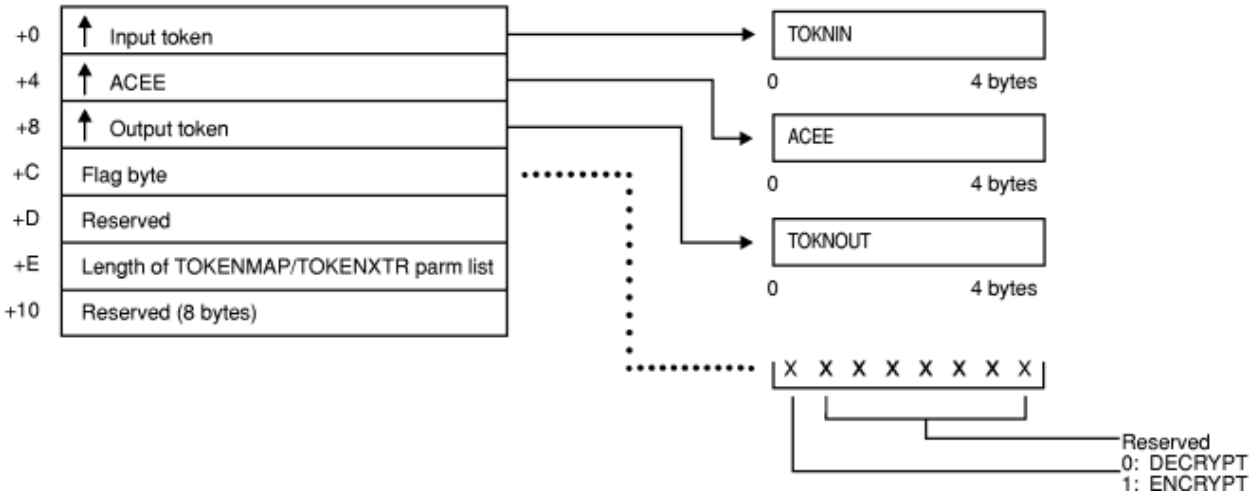


Figure 42. TOKENMAP/TOKENXTR service parameter list

VERIFY/VERIFYX service parameter list

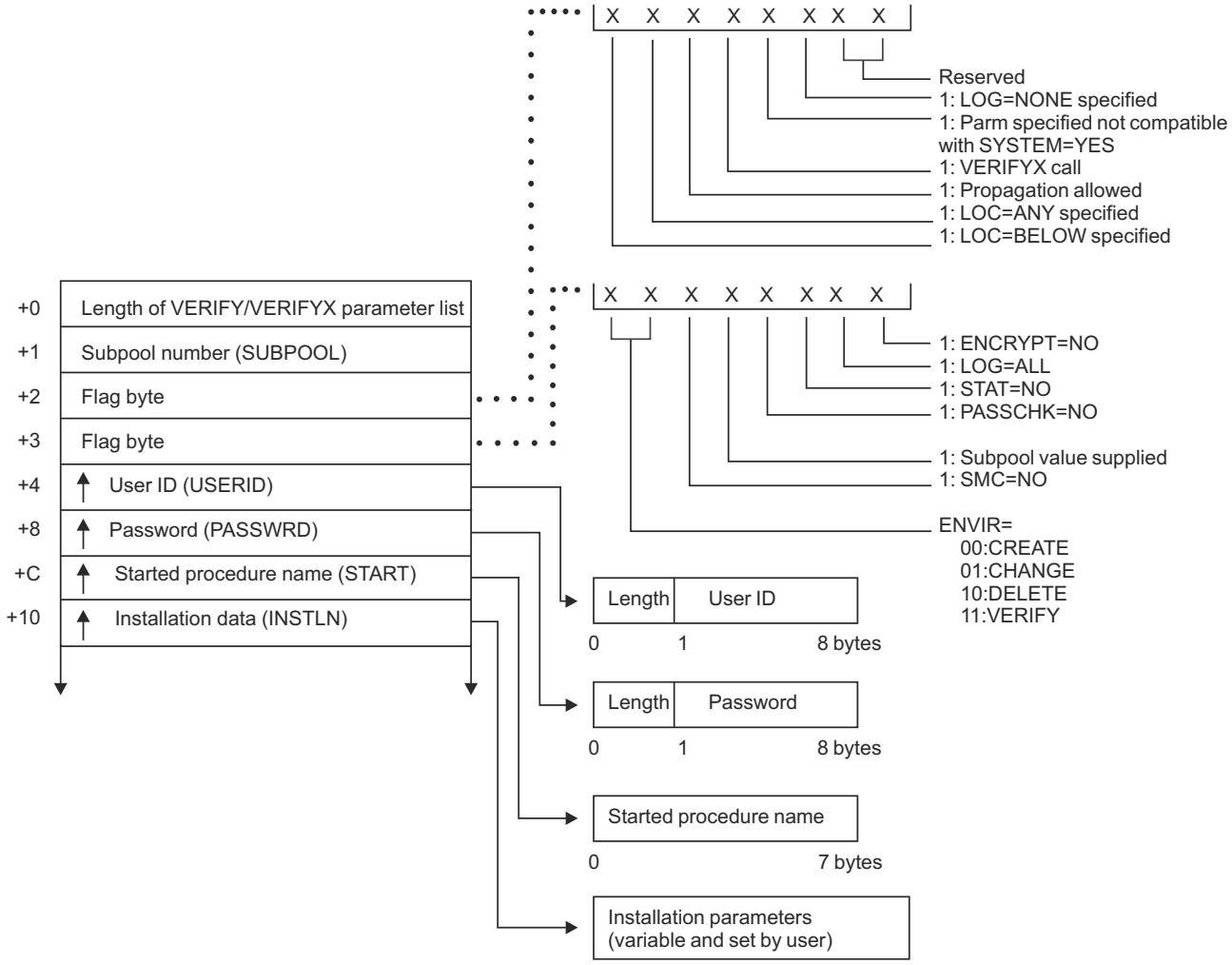


Figure 43. VERIFY/VERIFYX service parameter list (part 1 of 4)

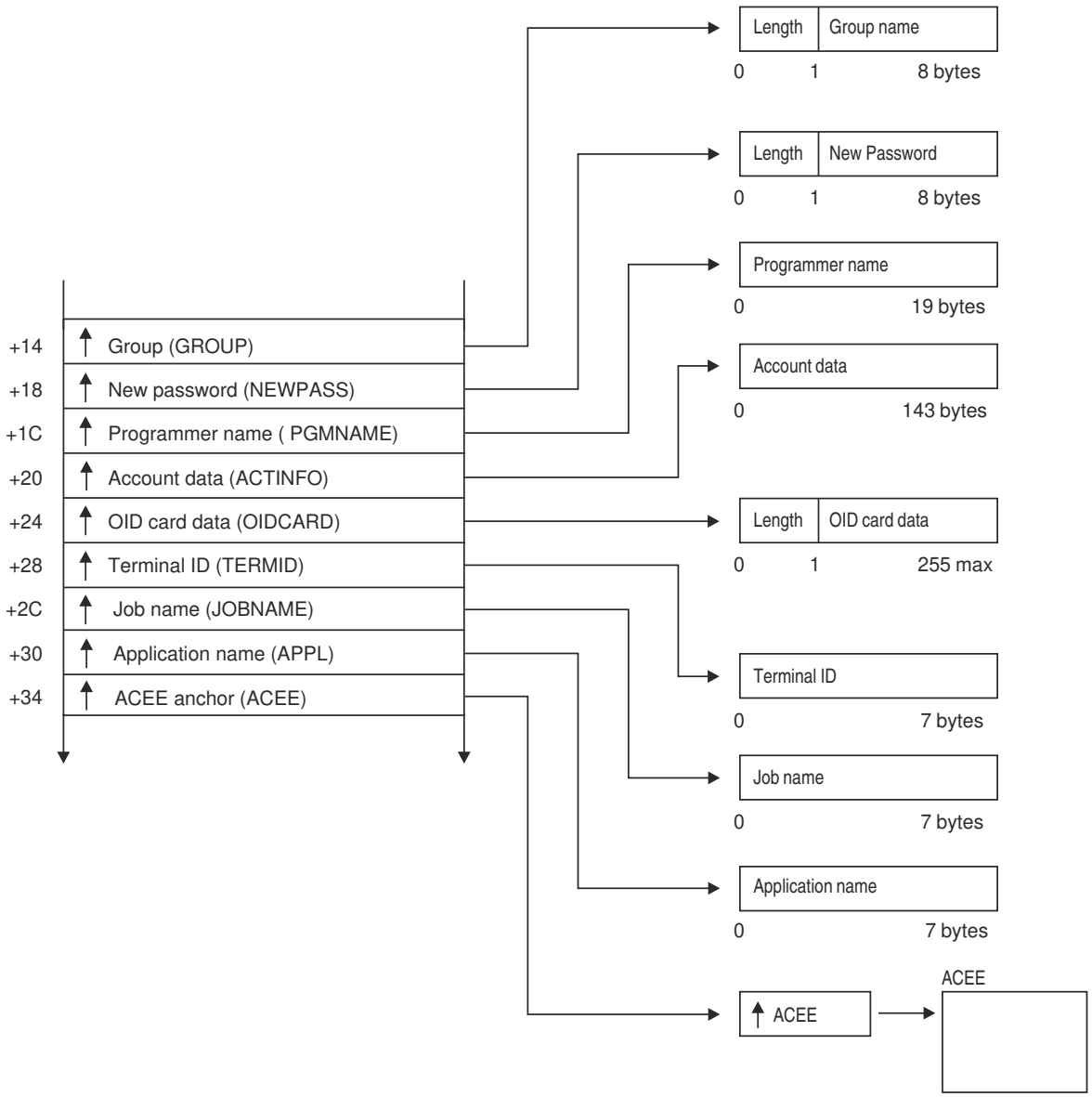


Figure 44. VERIFY/VERIFYX service parameter list (part 2 of 4)

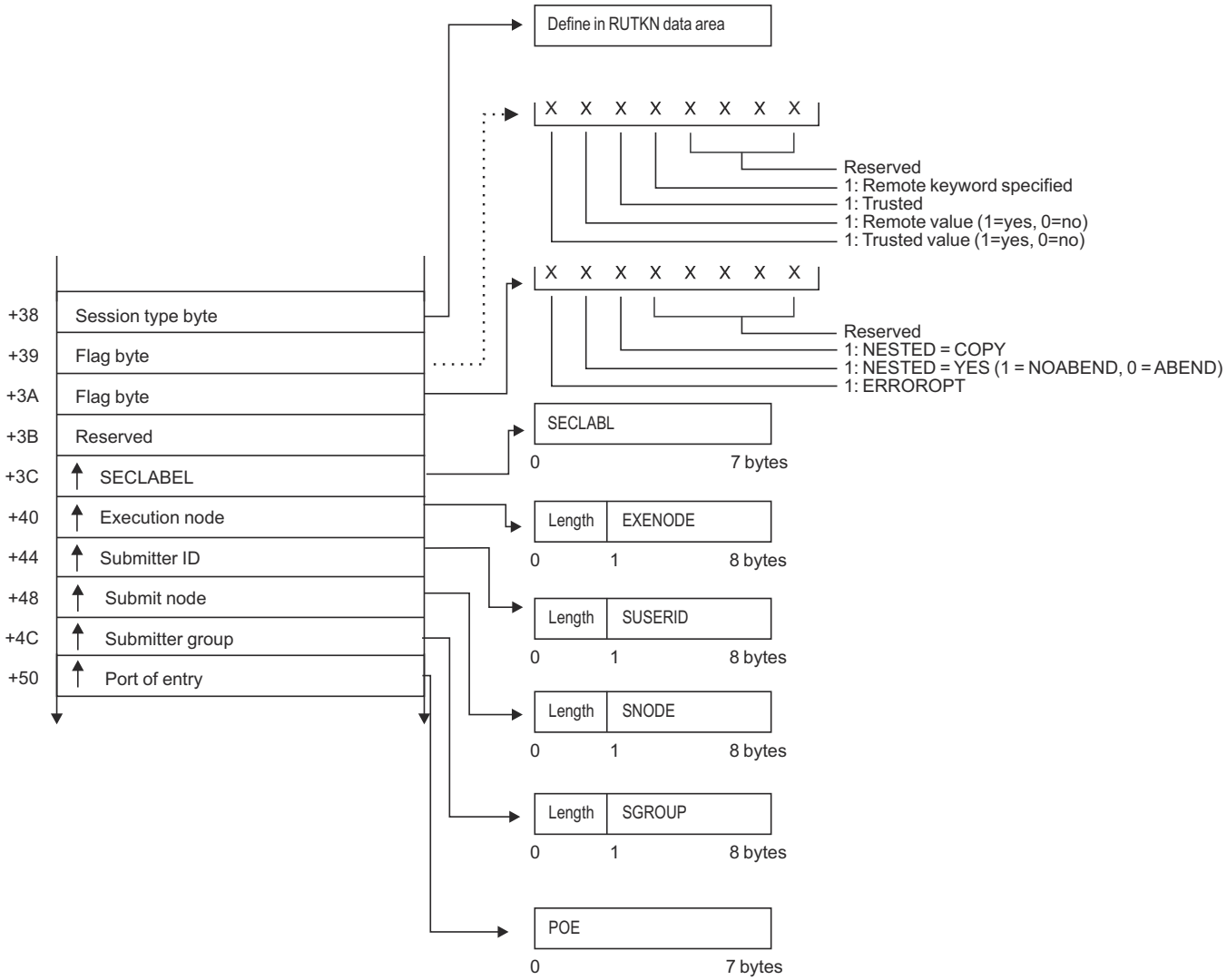


Figure 45. VERIFY/VERIFYX service parameter list (part 3 of 4)

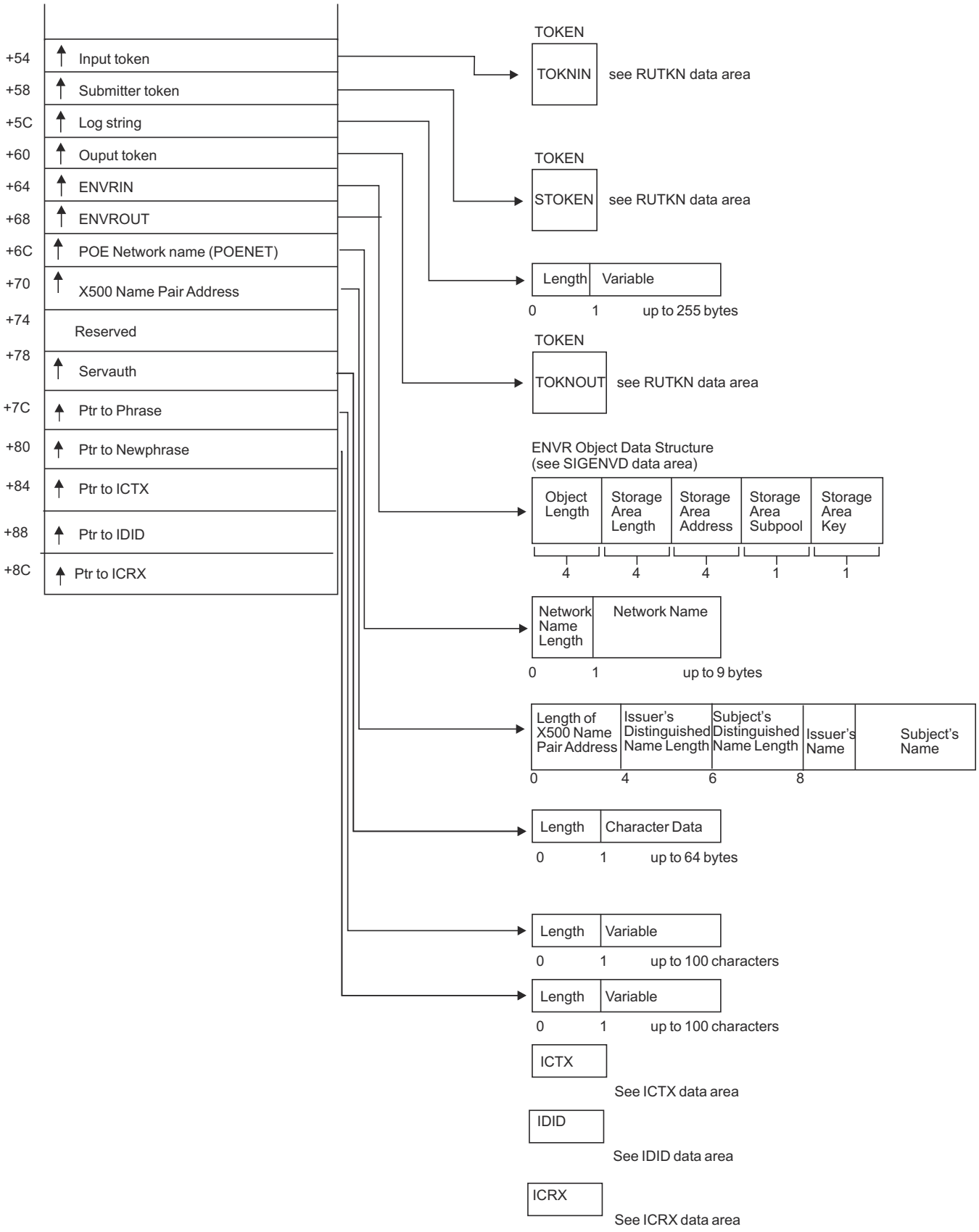


Figure 46. VERIFY/VERIFYX service parameter list (part 4 of 4)

ICHEINTY parameter list

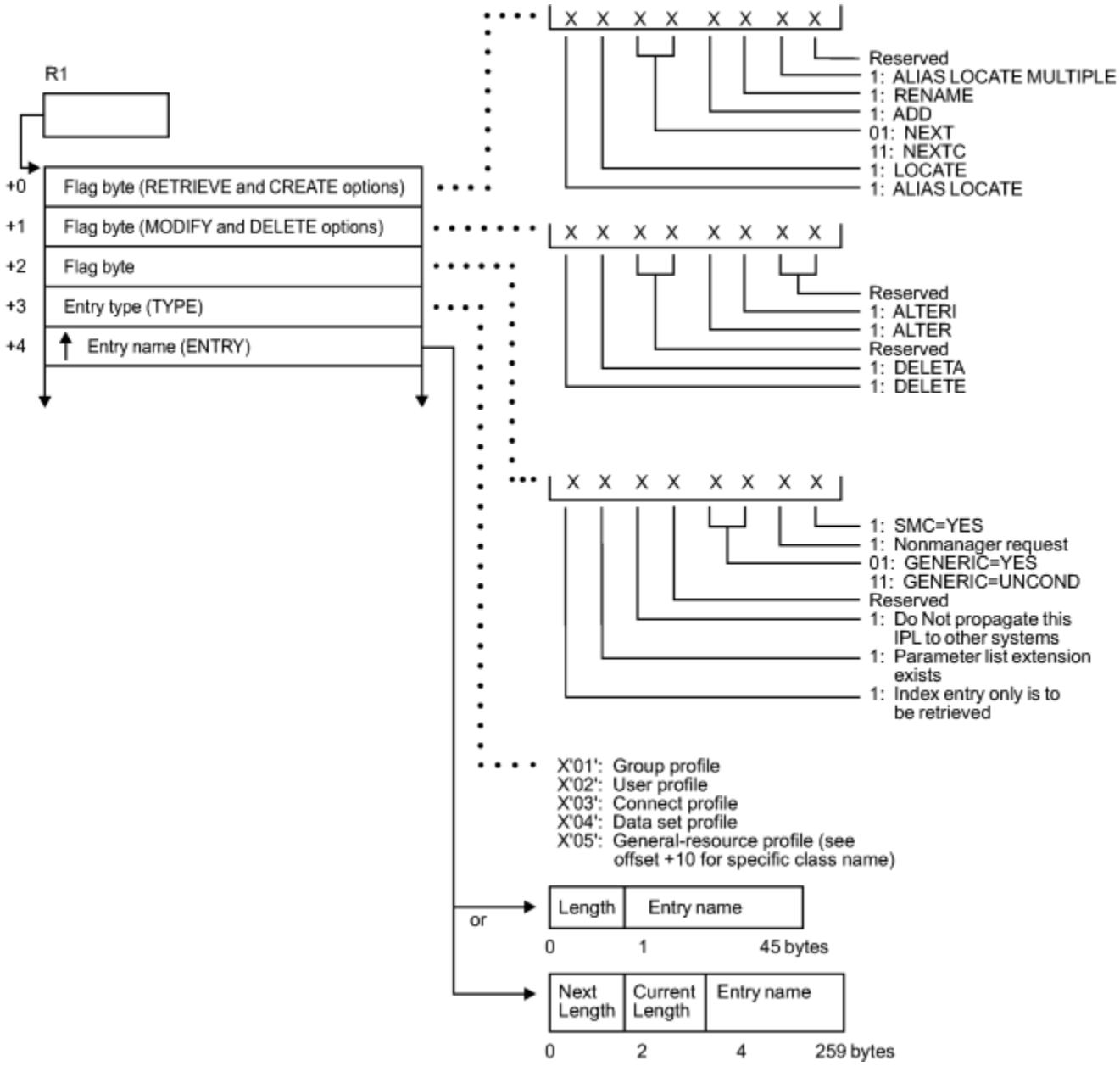
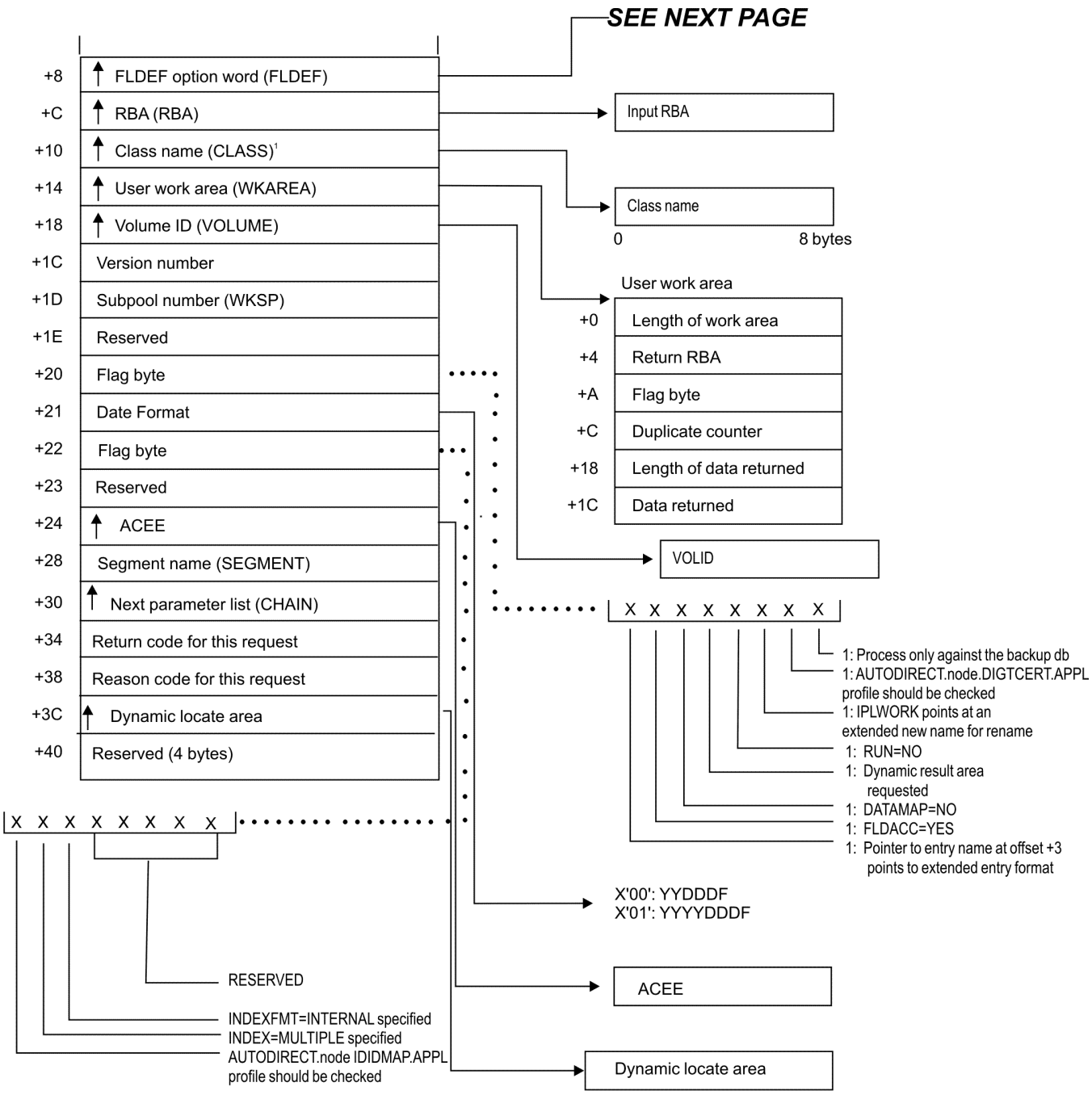


Figure 47. ICHEINTY parameter list (part 1 of 3)



[1] This field has a value only when the byte at +3 (entry type) has a value of X'05'.

Figure 48. ICHEINTY parameter list (part 2 of 3)

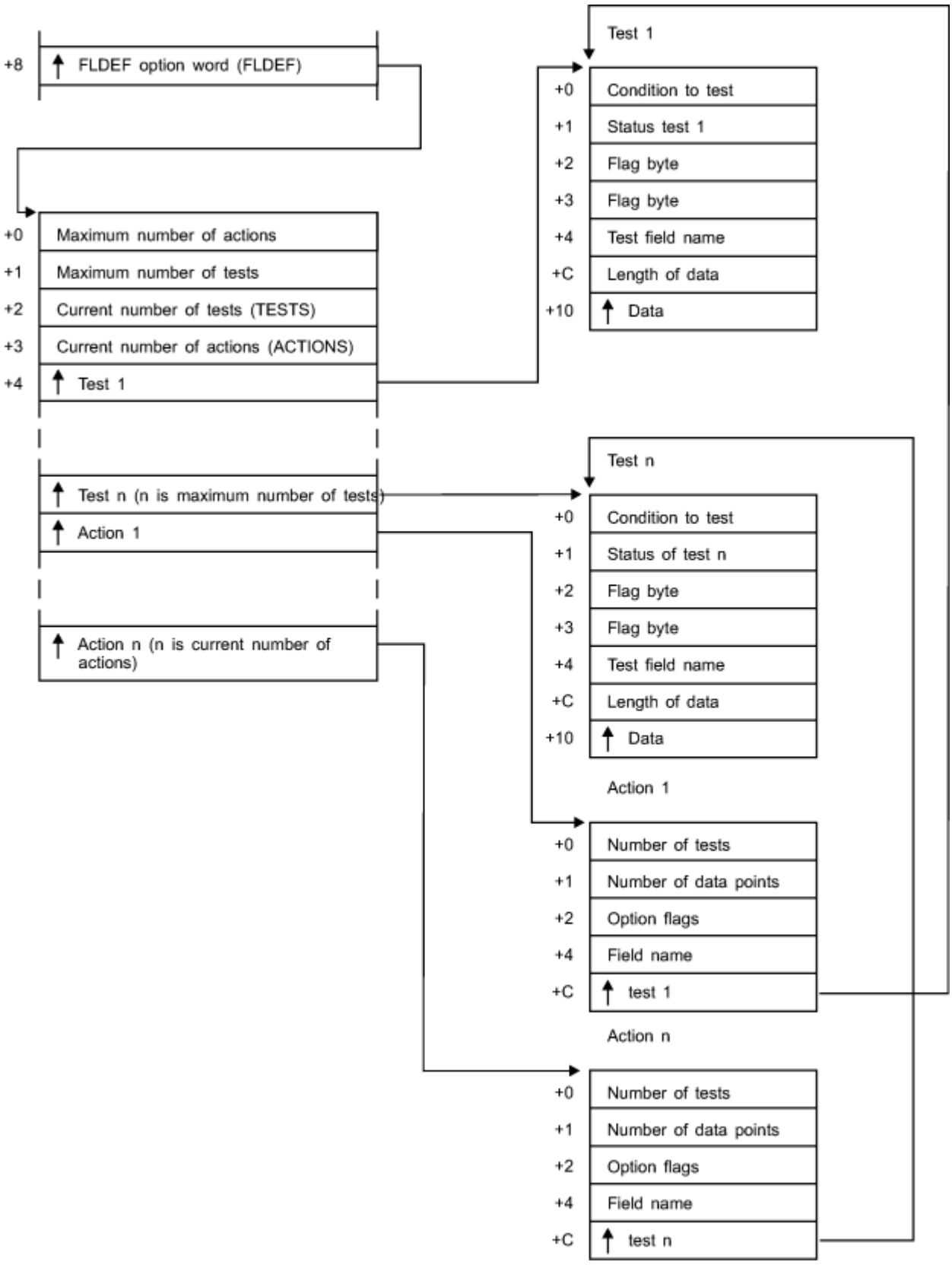


Figure 49. ICHEINTY parameter list (part 3 of 3)

Process flows for RACF

This provides process flows for:

- Commands
- Directed commands
- Macros
- Installation exits
- Utilities
- The report writer
- The SMF data unload utility
- Callable services

Process flow for RACF commands

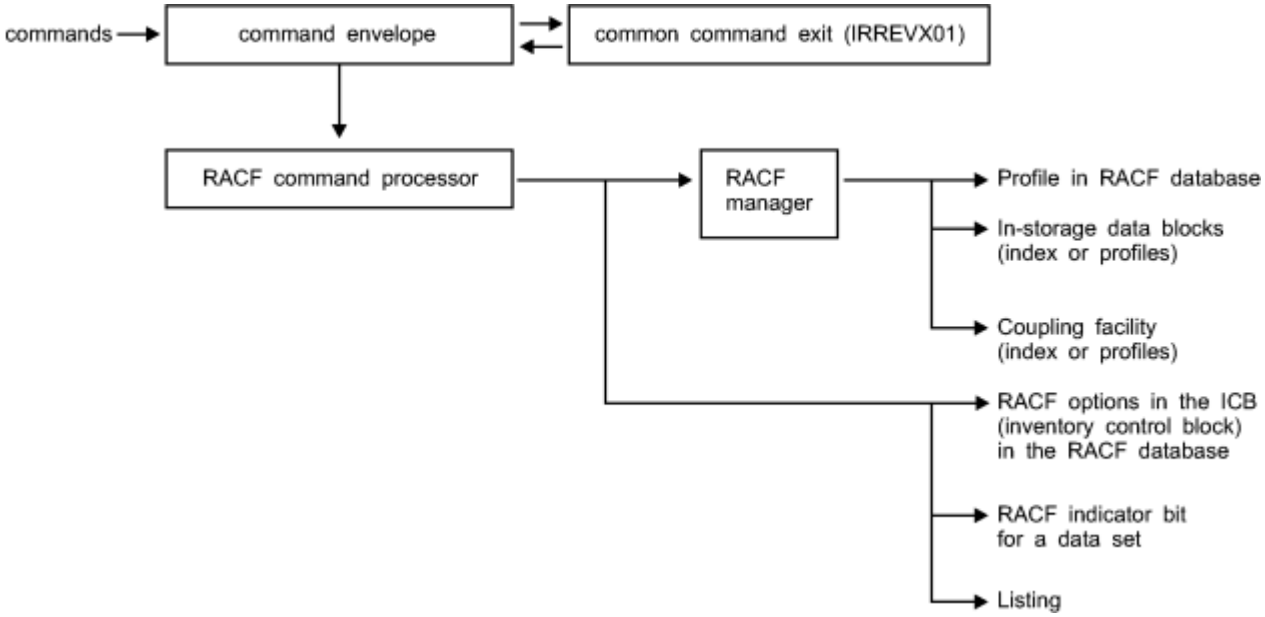


Figure 50. Process flow for RACF commands

Process flow for RACF directed commands

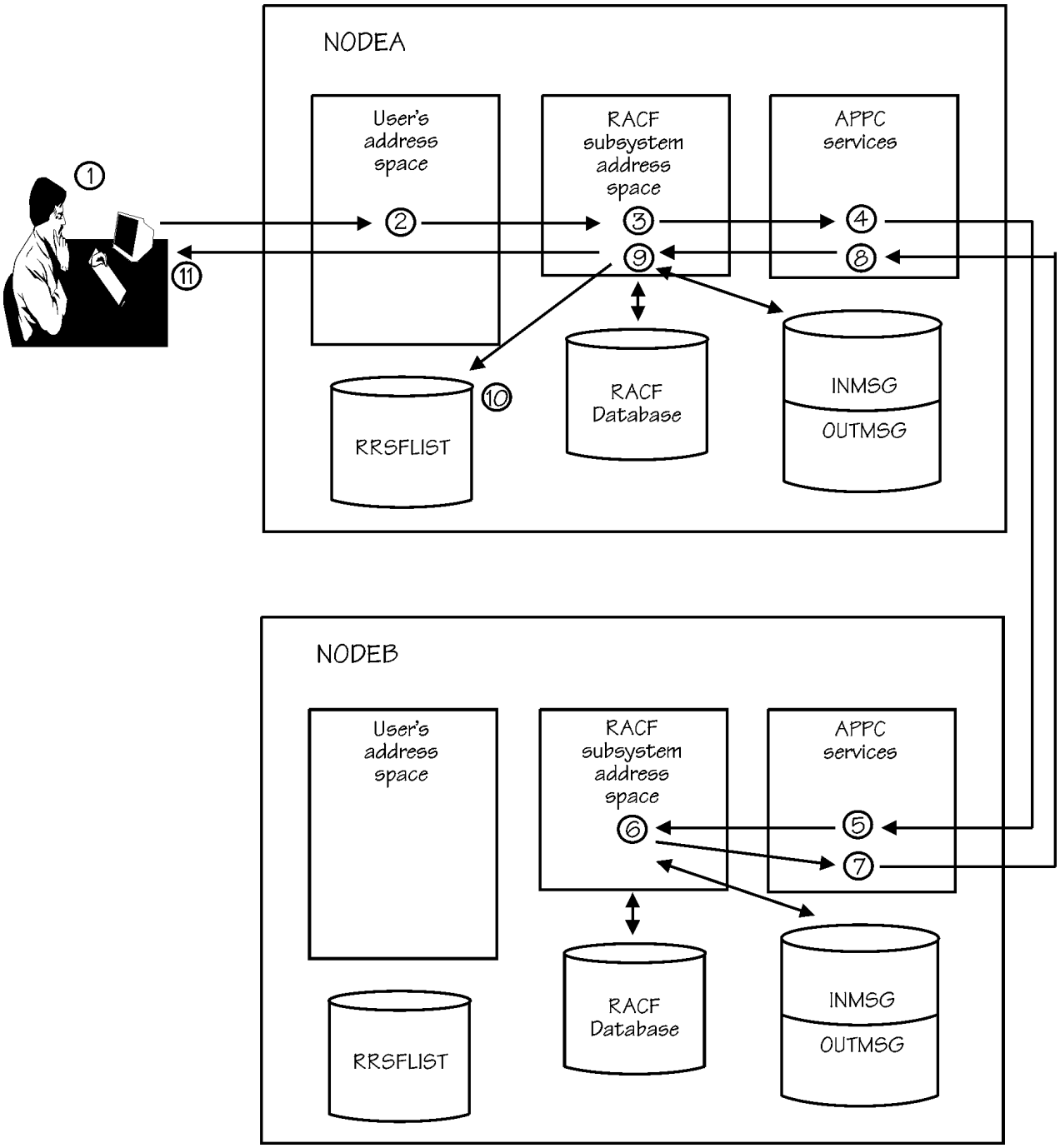


Figure 51. Process flow for RACF directed commands

Process flow for RACF macros

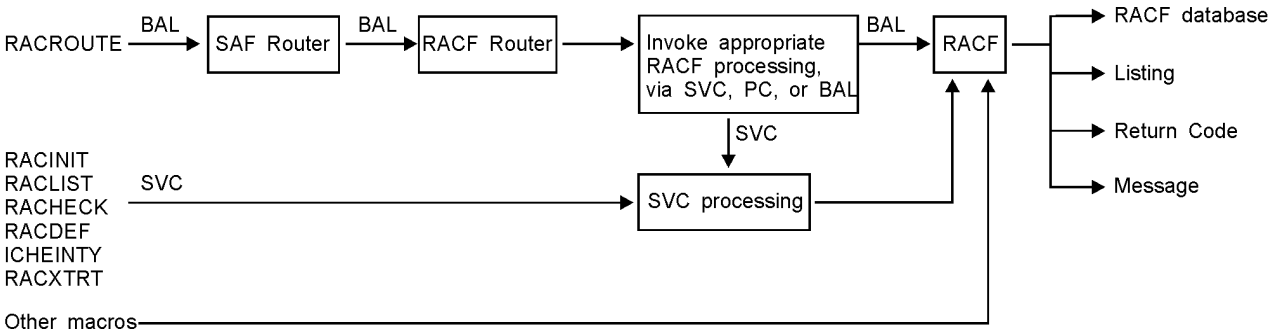


Figure 52. Process flow for RACF macros

Process flow for RACF installation exits

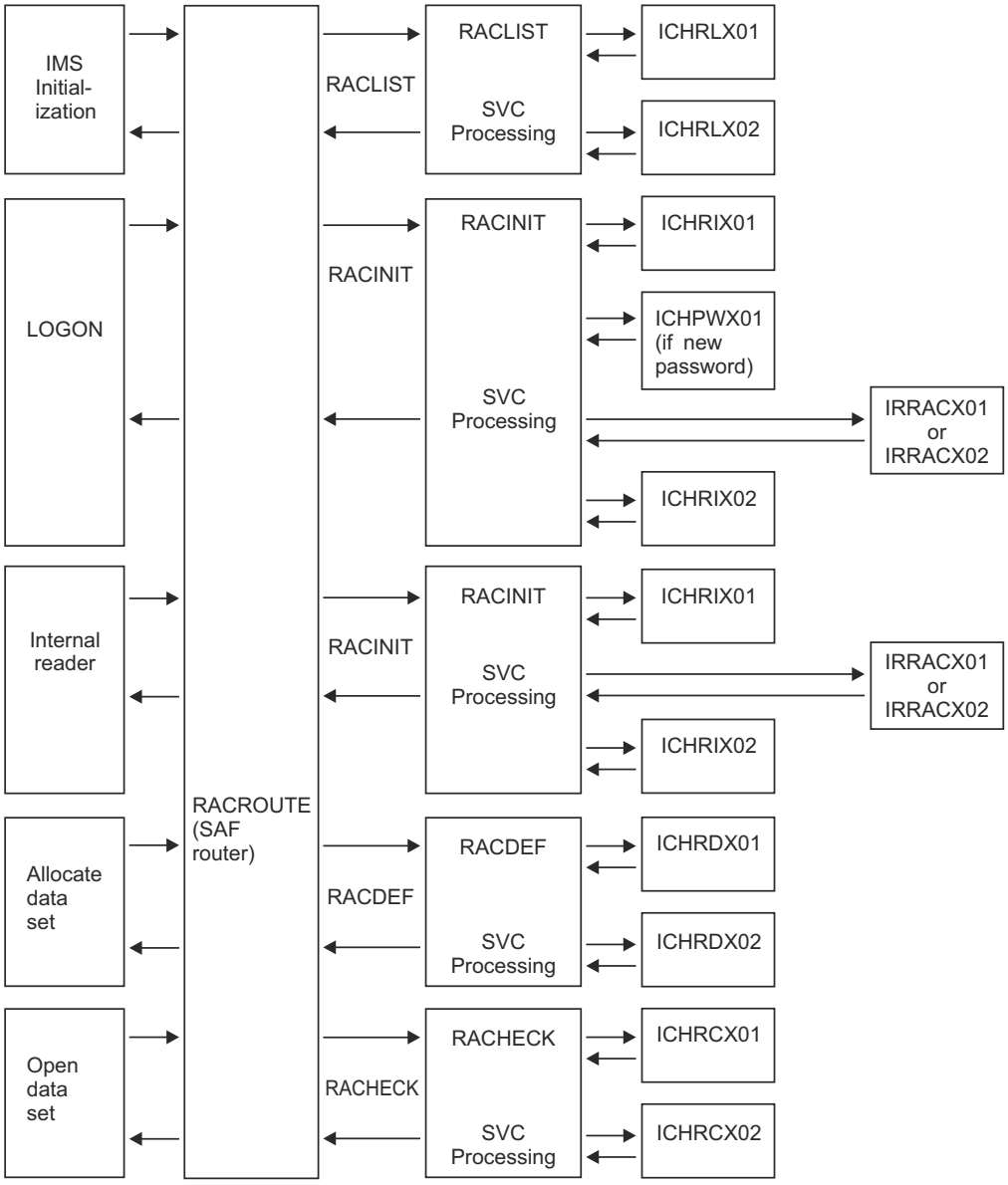


Figure 53. Process flow for RACF installation exits (part 1 of 2)

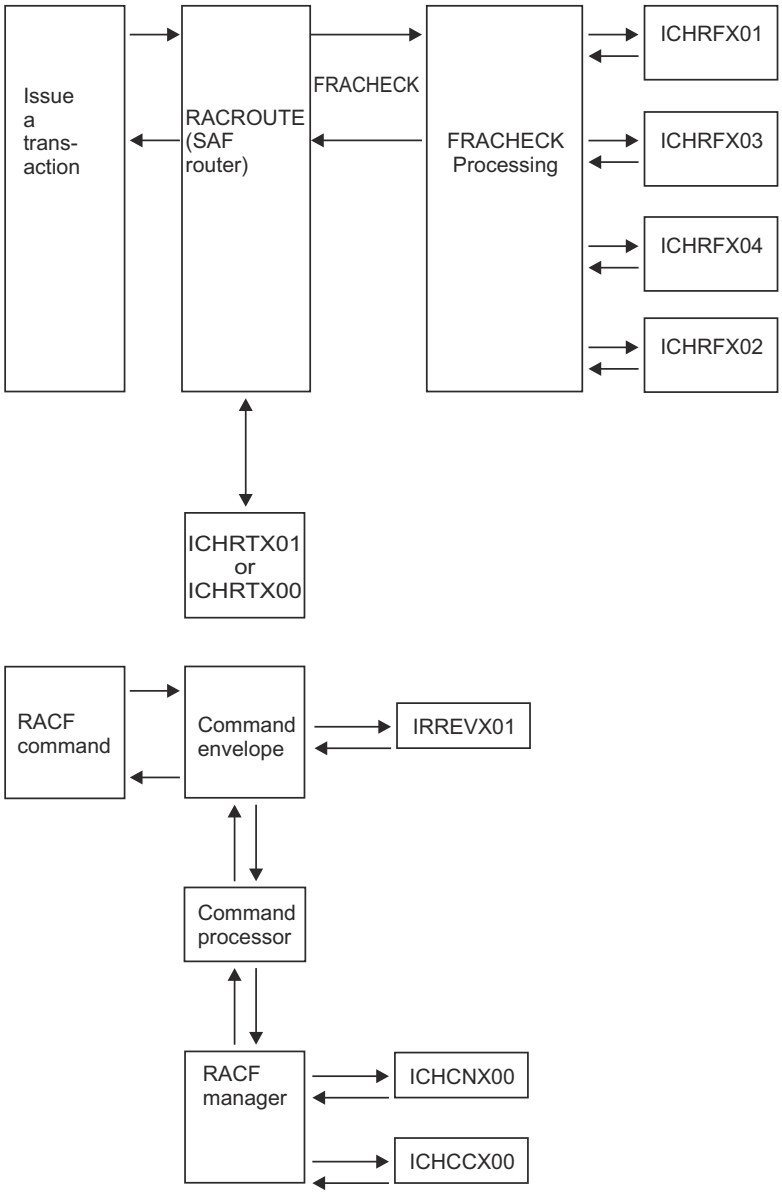


Figure 54. Process flow for RACF installation exits (part 2 of 2)

Process flow for RACF utilities

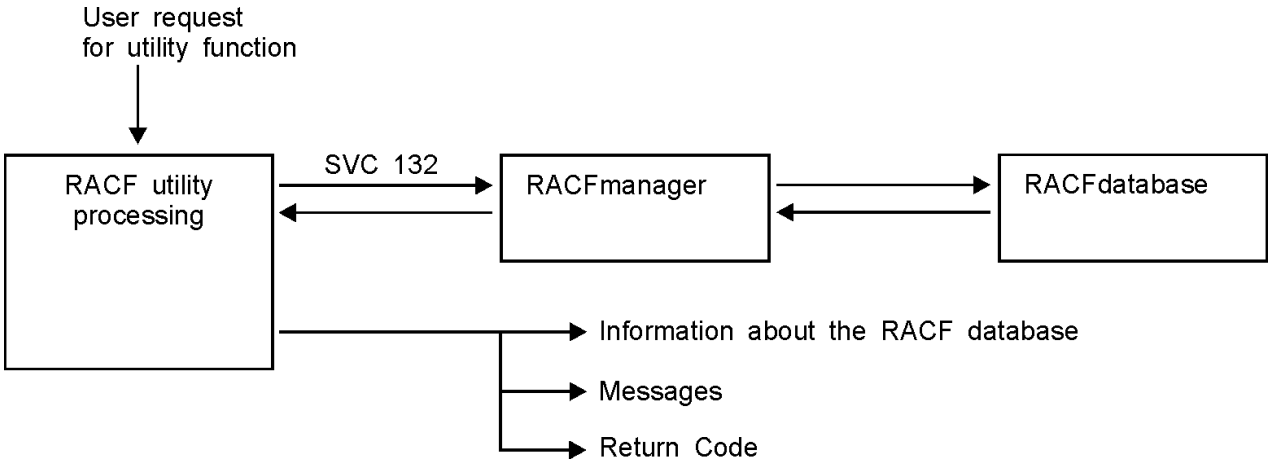


Figure 55. Process flow for RACF cross-reference and database unload utilities

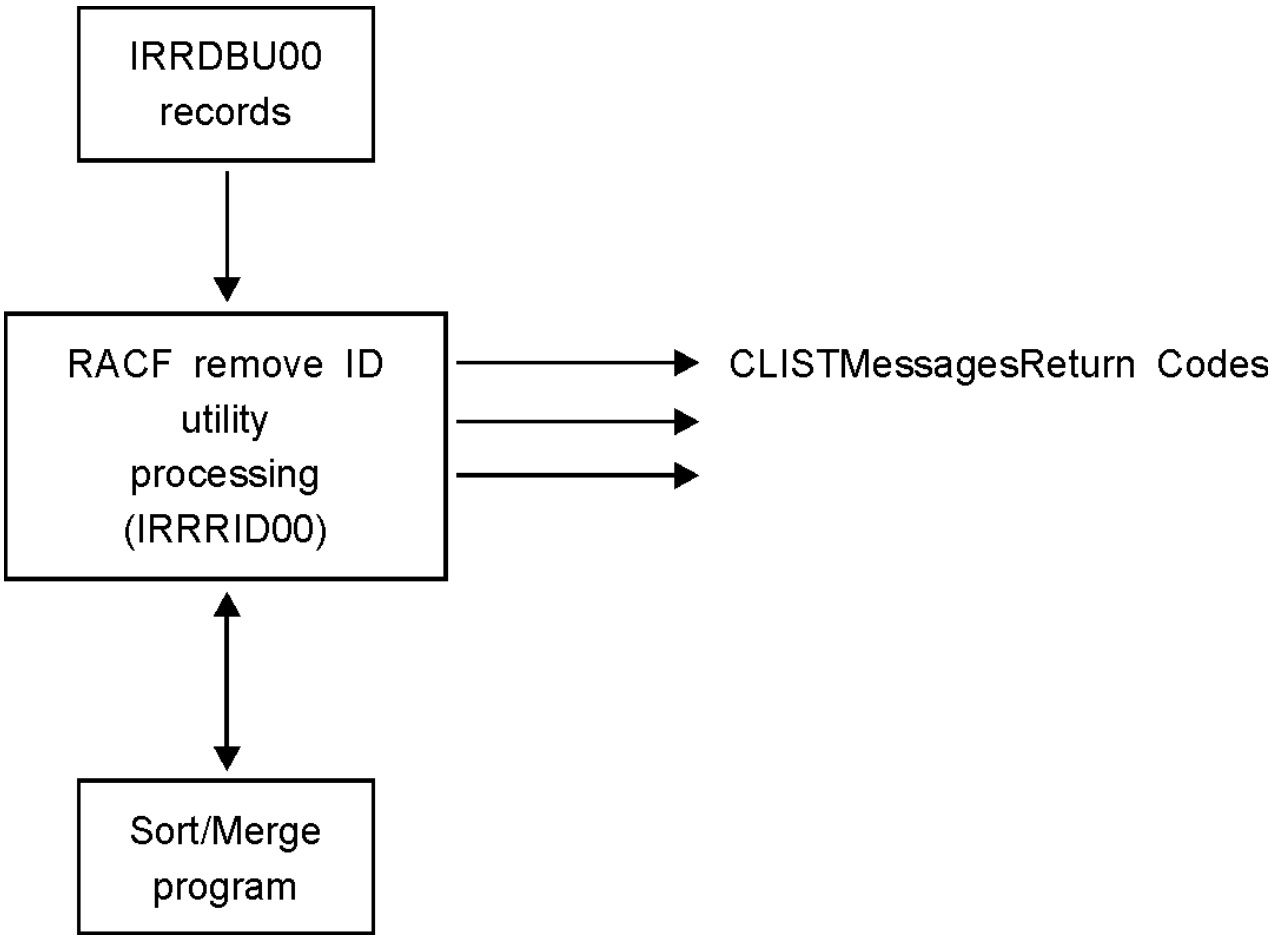


Figure 56. Process flow for RACF remove ID Utility

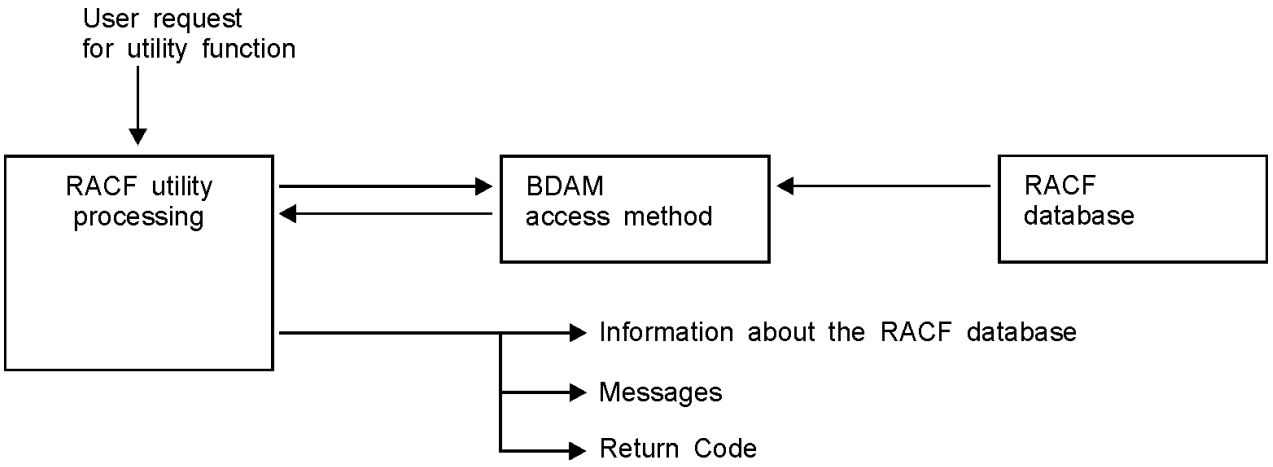


Figure 57. Process flow for other RACF Utilities

Process flow for the RACF report writer

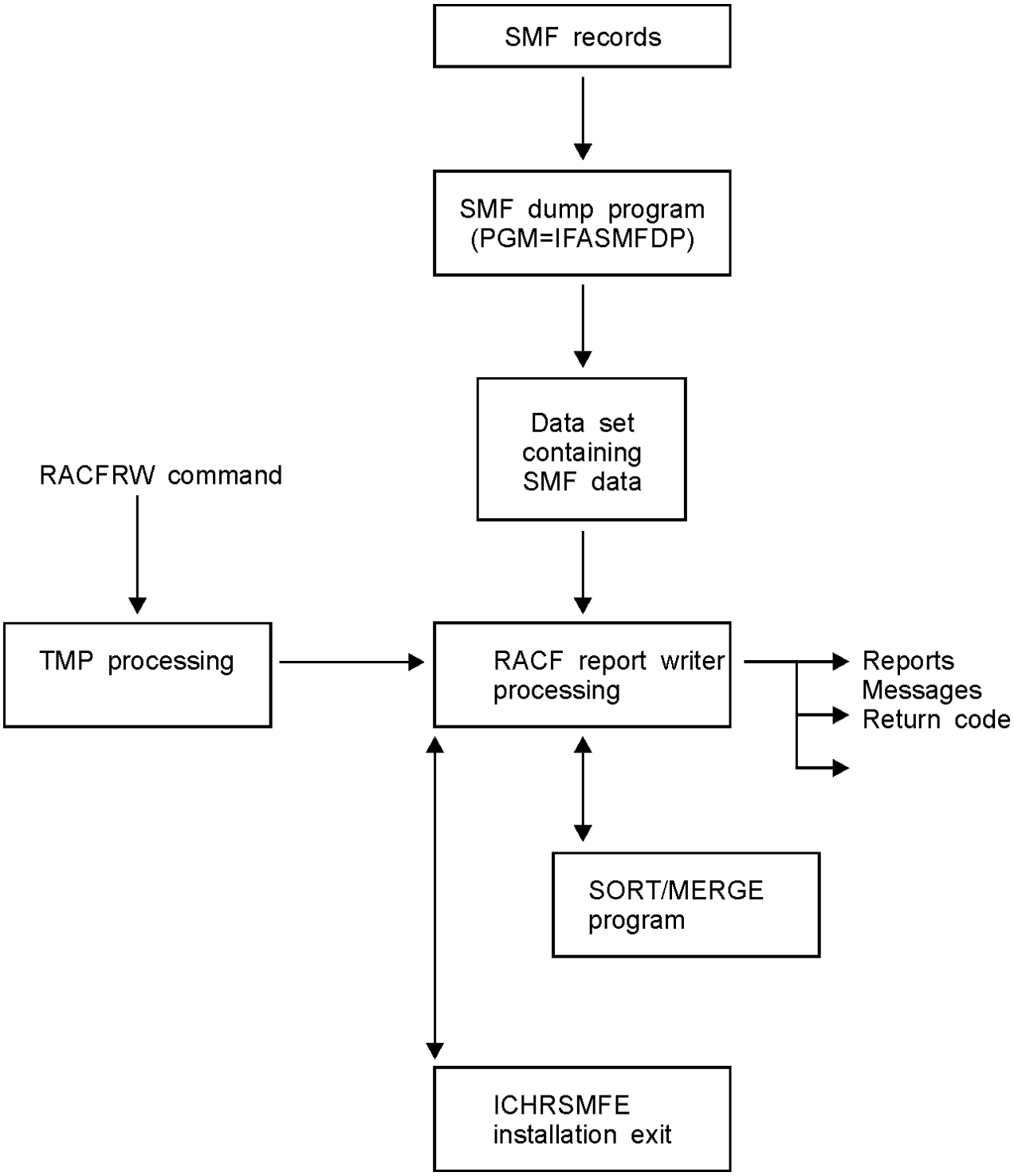


Figure 58. Process flow for the RACF report writer

Process flow for SMF data unload utility (IRRADU00)

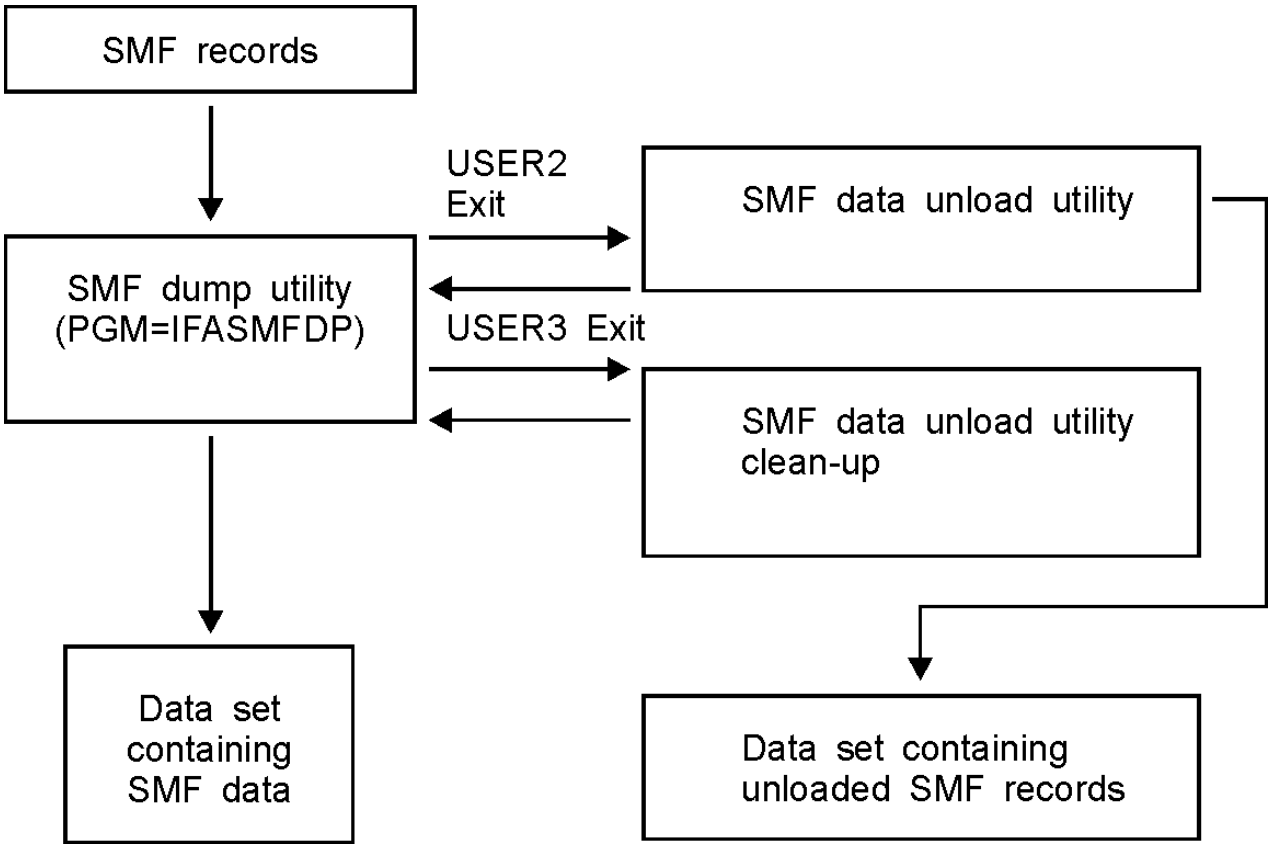


Figure 59. Process flow for SMF data unload utility

Process flow for RACF callable services

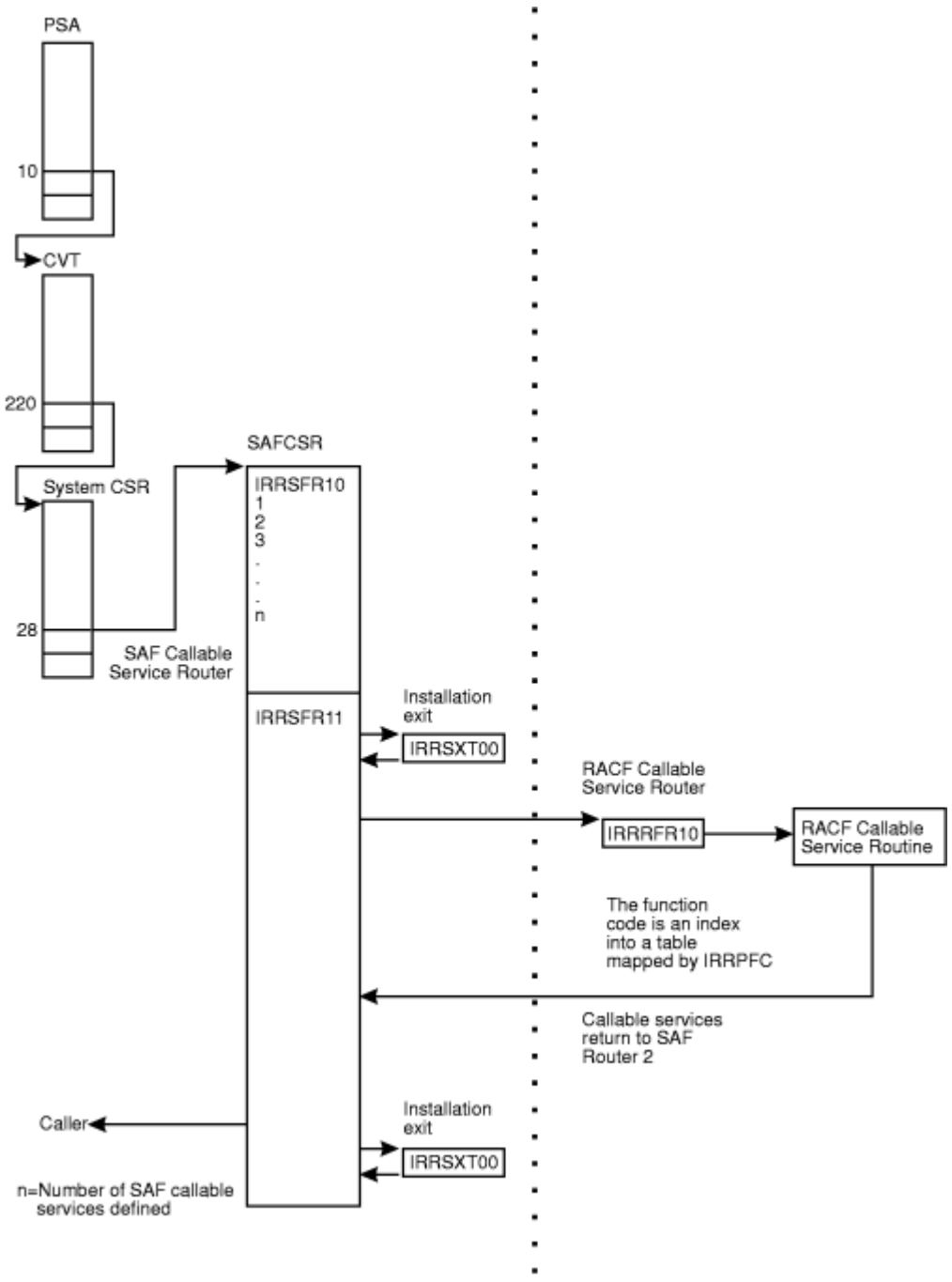


Figure 60. Process flow for RACF callable services

Error recording for callable services

Environmental errors must be reported to allow their correction. Some of the RACF callable services create symptom records for errors and store them in the LOGREC data set.

z/OS MVS Programming: Assembler Services Guide categorizes information contained in the symptom records as:

- Component data

  This identifies the application in which the error occurred (that is, the RACF subsystem address space).
- Primary SDB symptoms

This contains the primary string of problem symptoms. This data is used for duplicate problem recognition (see Table 37 on page 191).

- Secondary SDB symptoms

  This contains any additional diagnostic values saved at the time of the error. See Table 38 on page 191.

- Variable data

  This contains diagnostic data, such as portions of data areas or parameter lists pertinent to the error. See Table 39 on page 191.

Table 37 on page 191 shows the contents of the primary symptom string in the symptom record.

Table 37. Contents of primary symptom string for a callable service error

SDB key literal	Usage
RIDS/	Module CSECT name
RIDS/	Load module name (IRRRFR10)
PCSS/	System service (MVS or RACF macro) that failed
PRCS/	Return code from service
PRCS/	Reason code from service (if no reason code is available, this field contains a value of zero)

Table 38 on page 191 shows the contents of the secondary symptom string.

Table 38. Contents of secondary symptom string for a callable service error

SDB key literal	Usage
ADRS/	Offset of service that failed

Table 39 on page 191 shows the additional information that the symptom record contains.

Table 39. Additional LOGREC data for a callable service error

Field	Value
COMPDSC	RACF callable service
PROBLEM	Name of failing service
SERVLEV	Service level of module
COMPID	Component ID (5752XXH00)
COMPLEV	Component level (260)

Variable data recorded by RACF callable services

Each RACF callable service records additional variable data specific to the problem:

R_usermap (CSECT IRRRIM00)

- Error determining the RACF user ID associated with an application user identity
  - Service name: RACXTRT or ICHEINTY
  - Variable data: The string `'IRRRIM00_XXXX:'`, where XXXX is the procedure name from which error occurs
  - Additional variable data: application user identity and field names, such as SNAME or UNAME
- Error determining the application user identity associated with a RACF user ID
  - Service name: RACXTRT

  - Variable data: The string `'IRRRIM00_XXXX:'`, where XXXX is the procedure name from which error occurs
  - Additional variable data: RACF user ID and field names, such as SNAME or UNAME
- Figure 61 on page 192 shows an example of a symptom record when the user name is not found in an alias index entry but is found in the mapping profile.

```
TYPE:  SYMPTOM RECORD    REPORT:  SOFTWARE EDIT REPORT     DAY YEAR
                                  REPORT DATE:  032  00
SCP:   VS 2 REL 3                 ERROR DATE:    032  00
                       MODEL:  4381             HH  MM  SS.TH
                       SERIAL: 127685           TIME: 08:28:48.85
SEARCH ARGUMENT ABSTRACT:
            PIDS/5752XXH00 PRCS/0000000C PRCS/00000000 RIDS/IRRRIM00
            RIDS/IRRRFR10#L
SYSTEM ENVIRONMENT:
          CPU MODEL:  4381                     DATE:  032  00
          CPU SERIAL: 127685                   TIME:  08:28:48.85
          SYSTEM:     RACFR10                  BCP:   MVS
          RELEASE LEVEL OF SERVICE ROUTINE:      HBB7703
          SYSTEM DATA AT ARCHITECTURE LEVEL:     10
          COMPONENT DATA AT ARCHITECTURE LEVEL:  10
          SYSTEM DATA:  00000000 00000000        |........|
COMPONENT INFORMATION:
            COMPONENT ID:             5752XXH00
            COMPONENT RELEASE LEVEL:  7703
            SERVICE RELEASE LEVEL:    HRF7703
            DESCRIPTION OF FUNCTION:  RACF z/OS UNIX SYSTEM CALLABLE SERV.
            PROBLEM ID:               ICHEINTY

PRIMARY SYMPTOM STRING:
            PIDS/5752XXH00 PRCS/0000000C PRCS/00000000 RIDS/IRRRIM00
            RIDS/IRRRFR10#L
            SYMPTOM            SYMPTOM DATA     EXPLANATION
            ---------------    -------------    -----------
            PIDS/5752XXH00     5752XXH00        COMPONENT IDENTIFIER
            PRCS/0000000C      0000000C         RETURN CODE
            PRCS/00000000      00000000         RETURN CODE
            RIDS/IRRRIM00      IRRRIM00         ROUTINE IDENTIFIER
            RIDS/IRRRFR10#L    IRRRFR10#L       ROUTINE IDENTIFIER

SECONDARY SYMPTOM STRING:
            ADRS/0000140C
            SYMPTOM            SYMPTOM DATA    EXPLANATION
           ---------------    -------------    -----------
            ADRS/0000140C      0000140C        STORAGE ADDRESS

FREE FORMAT COMPONENT INFORMATION:
KEY = F000     LENGTH = 000100 (0064)
+000   C9D9D9D9   C9D4F0F0   6DC19389   81A26DD3       |IRRRIM00_ALIAS_L|
+010   96837A40   E2D5C1D4   C540D4E8   40D3D596       |OC: SNAME MY LNO|
+020   A385A240   C9958696   40404040   40404040       |TES INFO        |
+030 LENGTH(0048) ==> ALL BYTES CONTAIN X'40'.
+060    40404040                                      |                |

HEX DUMP OF RECORD:
HEADER
+000   4C831800   00000000   0000032F   08284885       |<C.............E|
+010   FF127685   43810000                            |    ...E.A..    |
```

Figure 61. An example of a symptom record when an SNAME is not found in the alias index entry but is found in the mapping profile (part 1 of 2)

```
SYMPTOM RECORD
+000   E2D9F4F3   F8F1F1F2   F7F6F8F5   FFFFBCF1      |SR4381127685...1|
+010   B388C6D9   4BC38185   40404040   40404040      |.HFR.CAE        |
+020   4040D9C1   C3C6D9F1   F040F5F7   F5F2C8C2      |RACFR10 5752HB  |
+030   C2F7F7F0   F3400080   00000000   00000000      |B7703 ..........|
+040   F1F00030   00640070   00490138   000E0181      |10.............A|
+050   0068018F   00000000   00000000   00000000      |................|
+060   00000000   00000000   00000000   00000000      |................|
+070   E2D9F2F1   F1F0F5F7   F5F2E7E7   C8F0F000      |SR21105752XXH00.|
+080   F7F7F0F3   C8D9C6F7   F7F0F340   00000000      |7703HRF7703 ....|
+090   00000000   00000000   00000000   D9C1C3C6      |............RACF|
+0A0   40D69785   95C58489   A3899695   40C38193      |z/OS UNIX SYSTEM|
+0B0   93818293   8540E285   99A54B40   00000000      |CALLABLE SERV. .|
+0C0   00000000   C9C3C8C5   C9D5E3E8   00000000      |....ICHEINTY....|
+0D0   00000000   066839B1   066839B1   066839BF      |................|
+0E0   066839BF   06683A27   06683A27   01000000      |................|
+0F0   06683830   00000000   00000000   F0F1F2F3      |............0123|
+100   F4F5F6F7   F8F9C1C2   C3C4C5C6   0049000E      |456789ABCDEF....|
+110   0068000F   00000000   F0F0F0F0   F1F4F0C3      |........0000140C|
+120   00000000   00000000   00000000   00000000      |................|
+130   00000000   00000000   D7C9C4E2   61F5F7F5      |........PIDS/575|
+140   F2E7E7C8   F0F040D7   D9C3E261   F0F0F0F0      |2XXH00 PRCS/0000|
+150   F0F0F0C3   40D7D9C3   E261F0F0   F0F0F0F0      |000C PRCS/000000|
+160   F0F040D9   C9C4E261   C9D9D9D9   C9D4F0F0      |00 RIDS/IRRRIM00|
+170   40D9C9C4   E261C9D9   D9D9C6D9   F1F07BD3      | RIDS/IRRRFR10#L|
+180   40C1C4D9   E261F0F0   F0F0F1F4   F0C340F0      | ADRS/0000140C 0|
+190   000064C9   D9D9D9C9   D4F0F06D   C1938981      |...IRRRIM00_ALIA|
+1A0   A26DD396   837A40E2   D5C1D4C5   40D4E840      |S_LOC: SNAME MY |
+1B0   D3D596A3   85A240C9   95869640   40404040      |LNOTES INFO     |
+1C0 LENGTH(0048) ==>ALL BYTES CONTAIN X'40'.
```

Figure 62. An example of a symptom record when an SNAME is not found in the alias index entry but is found in the mapping profile (part 2 of 2)

getGMAP (CSECT IRRRGM01)

- Error extracting the group name corresponding to the input group ID
  - Service name: RACXTRT or ICHEINTY
  - Variable data: The string 'IRRRGM01_XXXX: YYYY', where XXXX is the procedure name from which error occurs and YYYY is the GROUP mapping profile name for RACXTRT service or is the GROUP Alias name for ICHEINTY service
- Error extracting the group ID corresponding to the input group name
  - Service name: RACXTRT
  - Variable data: The string 'IRRRGM01_XXXX: YYYY', where XXXX is the procedure name from which error occurs and YYYY is the GROUP profile name

getGMAP/getUMAP/initUSP (CSECT IRRRUM02)

- Error locating or updating a profile during automatic UID/GID assignment
  - Service name: ICHEINTY
  - Variable data: The string 'IRRRUM02_XXXX: YYYY', where XXXX is the procedure name from which error occurs and YYYY is the ICHEINTY operation
  - Additional Variable data: class name and profile name
  - Primary symptom string return code: ICHEINTY return code
  - Secondary symptom string reason code: ICHEINTY reason code
- Error processing APPLDATA for the BPX.NEXT.USER profile in the FACILITY class
  - Service name: BPXNEXT
  - Variable data: The string 'IRRRUM02_XXXX: YYYY', where XXXX is the procedure name from which error occurs and YYYY is the profile being processed for automatic UID/GID assignment
  - Additional variable data: One of the following error descriptions:
    - Profile BPX.NEXT.USER ran out of UID values
    - Profile BPX.NEXT.USER ran out of GID values

getUMAP (CSECT IRRRUM01)

- Error extracting the user name corresponding to the input user ID
  - Service name: RACXTRT or ICHEINTY
  - Variable data: The string 'IRRRUM01_XXXX: YYYY', where XXXX is the procedure name from which error occurs and YYYY is the USER mapping profile name for RACXTRT service or is the USER application identity name for ICHEINTY service
- Error extracting the user ID corresponding to the input user name
  - Service name: RACXTRT
  - Variable data: The string 'IRRRUM01_XXXX: YYYY', where XXXX is the procedure name from which the error occurs and YYYY is the USER profile name
- Figure 63 on page 194 shows an example of a symptom record when the user name is not found in an alias index entry but is found in the mapping profile.

```
TYPE:  SYMPTOM RECORD        REPORT:  SOFTWARE EDIT REPORT        DAY YEAR
                                      REPORT DATE: 032  00
SCP:   VS 2 REL 3                     ERROR DATE: 032  00
                    MODEL:  4381               HH  MM  SS.TH
                    SERIAL: 127685   TIME: 09:16:23.25
SEARCH ARGUMENT ABSTRACT:
           PIDS/5752XXH00 PRCS/0000000C PRCS/00000000 RIDS/IRRRUM01
           RIDS/IRRRFR10#L
SYSTEM ENVIRONMENT:
           CPU MODEL:  4381              DATE:  032  00
           CPU SERIAL: 127685            TIME:   09:16:23.25
           SYSTEM:  RACFR10              BCP:    MVS
           RELEASE LEVEL OF SERVICE ROUTINE:      HBB7703
           SYSTEM DATA AT ARCHITECTURE LEVEL:     10
           COMPONENT DATA AT ARCHITECTURE LEVEL:  10
           SYSTEM DATA:  00000000 00000000         |........|
COMPONENT INFORMATION:
           COMPONENT ID:            5752XXH00
           COMPONENT RELEASE LEVEL:  7703
           SERVICE RELEASE LEVEL:    HRF7703
           DESCRIPTION OF FUNCTION:  RACF z/OS UNIX SYSTEM CALLABLE SERV.
           PROBLEM ID:               ICHEINTY
PRIMARY SYMPTOM STRING:
           PIDS/5752XXH00 PRCS/0000000C PRCS/00000000 RIDS/IRRRUM01
           RIDS/IRRRFR10#L

           SYMPTOM              SYMPTOM DATA     EXPLANATION
           --------------       ---------------  -------------
           PIDS/5752XXH00       5752XXH00        COMPONENT IDENTIFIER
           PRCS/0000000C        0000000C         RETURN CODE
           PRCS/00000000        00000000         RETURN CODE
           RIDS/IRRRUM01        IRRRUM01         ROUTINE IDENTIFIER
           RIDS/IRRRFR10#L      IRRRFR10#L       ROUTINE IDENTIFIER

SECONDARY SYMPTOM STRING:
           ADRS/00000BAA

           SYMPTOM              SYMPTOM DATA     EXPLANATION
           ---------------      -------------    -----------
           ADRS/00000BAA        00000BAA         STORAGE ADDRESS
FREE FORMAT COMPONENT INFORMATION:
KEY = F000     LENGTH = 000100 (0064)
+000    C9D9D9D9    E4D4F0F1    6DC1D3C9    C1E26DE4    |IRRRUM01_ALIAS_U|
+010    D3D6C37A    40F1F2F3    F4F5F6F7     F8404040   |LOC: 12345678   |
+020 LENGTH(0064) ==> ALL BYTES CONTAIN X'40'.
+060    40404040                                       |                |

HEX DUMP OF RECORD:
HEADER
+000    4C831800    00000000    0000032F    09162325    |<C.............|
+010    FF127685    43810000                            |...E.A..        |
```

Figure 63. An example of a symptom record when a UID is not found in the alias index entry but is found in the mapping profile (part 1 of 2)

```
SYMPTOM RECORD
+000    E2D9F4F3    F8F1F1F2    F7F6F8F5    FFFFBCF1    |SR4381127685...1|
+010    B388D17B    767DA544    40404040    40404040    |.HJ#.'V.        |
+020    4040D9C1    C3C6D9F1    F040F5F7    F5F2C8C2    |RACFR10 5752HB  |
+030    C2F7F7F0    F3400080    00000000    00000000    |B7703 ..........|
+040    F1F00030    00640070    00490138    000E0181    |10.............A|
+050    0068018F    00000000    00000000    00000000    |................|
+060    00000000    00000000    00000000    00000000    |................|
+070    E2D9F2F1    F1F0F5F7    F5F2E7E7    C8F0F000    |SR21105752XXH00.|
+080    F7F7F0F3    C8D9C6F7    F7F0F340    00000000    |7703HRF7703 ....|
+090    00000000    00000000    00000000    D9C1C3C6    |............RACF|
+0A0    40D69785    95C58489    A3899695    40C38193    |z/OS UNIX SYSTEM|
+0B0    93818293    8540E285    99A54B40    00000000    |CALLABLE SERV. .|
+0C0    00000000    C9C3C8C5    C9D5E3E8    00000000    |....ICHEINTY....|
+0D0    00000000    06694591    06694591    0669459F    |.......J...J....|
+0E0    0669459F    06694607    06694607    01000000    |................|
+0F0    06694410    00000000    00000000    F0F1F2F3    |............0123|
+100    F4F5F6F7    F8F9C1C2    C3C4C5C6    0049000E    |456789ABCDEF....|
+110    0068000F    00000000    F0F0F0F0    F0C2C1C1    |........00000BAA|
+120    00000000    00000000    00000000    00000000    |................|
+130    00000000    00000000    D7C9C4E2    61F5F7F5    |........PIDS/575|
+140    F2E7E7C8    F0F040D7    D9C3E261    F0F0F0F0    |2XXH00 PRCS/0000|
+150    F0F0F0C3    40D7D9C3    E261F0F0    F0F0F0F0    |000C PRCS/000000|
+160    F0F040D9    C9C4E261    C9D9D9D9    E4D4F0F1    |00 RIDS/IRRRUM01|
+170    40D9C9C4    E261C9D9    D9D9C6D9    F1F07BD3    | RIDS/IRRRFR10#L|
+180    40C1C4D9    E261F0F0    F0F0F0C2    C1C140F0    | ADRS/00000BAA 0|
+190    000064C9    D9D9D9E4    D4F0F16D    C1D3C9C1    |...IRRRUM01_ALIA|
+1A0    E26DE4D3    D6C37A40    F1F2F3F4    F5F6F7F8    |S_ULOC: 12345678|
+1B0 LENGTH(0064) ===> ALL BYTES CONTAIN X'40'.
```

Figure 64. An example of a symptom record when a UID is not found in the alias index entry but is found in the mapping profile (part 2 of 2)

InitACEE (CSECT IRRRIA00)

- Certificate decoding problem
  - Service name: DECX509B
  - Variable data: The following 6 fullwords:
    1. Address of the certificate passed to InitACEE
    2. Length of the certificate passed to InitACEE
    3. Address of the byte found to be in error
    4. Internal diagnostic word
    5. Number of certificate bytes recorded in the next 4 variable data sections
    6. Number of certificate bytes skipped at the beginning of the certificate when recording
  - Additional variable data: up to 1024 bytes of the certificate
- Error in certificate profile (user ID not valid)
  - Service name: GETUSERI
  - Variable data: The string 'Bad DIGTCERT Class Profile'
  - Additional variable data: The name of the DIGTCERT class profile found to be in error
- Error in certificate profile (user ID not defined to RACF)
  - Service name: RACINIT
  - Variable data: The string 'Bad DIGTCERT Class Profile'
  - Additional variable data: The name of the DIGTCERT class profile found to be in error
- Unexpected return code from R_cacheserv SAF callable service
  - Service name: CACHSERV
  - Variable Data: The 8 byte user ID input to R_cacheserv
  - Primary symptom string return code: R_cacheserv return code
  - Primary symptom string return code: R_cacheserv reason code

- Unexpected return code from RACROUTE REQUEST=VERIFY
  - Service name: RACINIT
  - Variable data: The nine character user ID being verified (1 byte length followed by up to 8 characters for the user ID itself)
- Problem with IDID user mapping
  - Service name: IRRDID2U
  - Variable data: Includes return code and IDID

InitACEE (CSECT IRRRIA02)

- Certificate decoding problem
  - Service name: DECX509B
  - Variable data: The following 6 fullwords:
    1. Address of the certificate passed to InitACEE
    2. Length of the certificate passed to InitACEE
    3. Address of the byte found to be in error
    4. Internal diagnostic word
    5. Amount of certificate data recorded
    6. Amount of certificate data truncated
  - Additional variable data: up to 1024 bytes of the certificate
- Error retrieving the CERTAUTH certificate repeat group from the `irrcerta` profile.
  - Service name: RACXTRT
  - Variable data: The string 'Couldn't extract the CERTAUTH anchor USER profile, `irrcerta`'.
  - Additional variable data: none
- Error extracting the private key size (CERTPRVS) from a DIGTCERT profile.
  - Service name: RACXTRT
  - Variable data: The string 'Bad CERTAUTH class profile:'*label-value*
  - Additional variable data: The profile name

InitACEE (CSECT IRRRIA05)

- Error extracting the trust status (UACC) from a DIGTCERT profile
  - Service name: RACEXTR
  - Variable data: The string 'Bad DIGTCERT Class Profile'
  - Additional variable data: The profile name
- Error extracting date fields CERTSTRT, CERTEND from a DIGTCERT profile
  - Service name: RACEXTR
  - Variable data: The string 'Bad DIGTCERT Class Profile'
  - Additional variable data: The profile name
- Error extracting the CERTAUTH anchor in the USER profile.
  - Service name: RACEXTR
  - Variable data: The string 'Bad USER class profile — `irrcerta`'

R_cacheserv (CSECT IRRRCH12)

- Physical error accessing the CACHECLS profile on the RACF database
  - Service name: ICHEINTY

  - Variable data: The string "LOCATE CACHECLS profile xxxx" where xxxx is the 16 byte profile name.
- Logical error with the contents of the CACHECLS profile read from the RACF database
  - Service name: Database (not a real Service, rather an indicator something was wrong with the profile read from the RACF database)
  - Variable data: The string "CACHECLS profile xxxx not valid:" where xxxx is the 16 byte profile name. Following that string is one of the following reasons the profile was determined not to be valid.
    - missing profile
    - profile not valid
    - timestamp is not correct
    - data space offset is not correct
    - data space size is not correct
    - data space count is not correct
    - cache version is not correct

R_cacheserv (CSECT IRRRCH13)

- Physical error accessing the CACHECLS profile on the RACF database
  - Service name: ICHEINTY
  - Variable data: The string "{LOCATE | ALTER | ADD} CACHECLS profile xxxx" where xxxx is the 16 byte profile name.

R_cacheserv (CSECT IRRRCH14)

- Physical error accessing the CACHECLS profile on the RACF database
  - Service name: ICHEINTY
  - Variable data: The string "{NEXT | DELETE} CACHECLS profile xxxx" where xxxx is the 16 byte profile name.

R_cacheserv (all CSECTs IRRRCHQx in IRRRCH16)

- Error occurred in data space layer
  - Service name: DISPLAYER
  - Variable data: xxxxxxxx yyyyyyyy zzzzzzzz aaaaaaaa bbbbbbbb: cccccccc
    - xxxxxxxx = Internal subroutine identifier

      The following are possible values for xxxxxxxx. These identify the failing data space layer subroutine:

Table 40. Failing data space layer for an internal subroutine identifier

xxxxxxxx	**Data space layer call**
8	dspCreate
24	dspDelete
28	dspConnect
32	dspExtend
36	dspExpose
40	dspHide
44	addAlet

Table 40. Failing data space layer for an internal subroutine identifier (continued)	
xxxxxxxx	**Data space layer call**
48	delAlet
52	dspAdd
60	dspDisconnect
64	dasENQDS
68	dasDEQDs
76	dasSetVer
80	dasGetVer
84	dasCopyDS

- yyyyyyyy = Failing system service

  The following are possible values for yyyyyyyy. These identify the failing system service or internal request:

Table 41. Failing system service for a failing system service identifier	
yyyyyyyy	**Failing system service**
0	None
4	TCBTOKEN
8	DSPSERV CREATE
12	ALESERV ADD
16	DSPSERV DELETE
20	Data space does not exist
24	Data space header eye-catcher bad
28	Irrrch10 not found
32	IEANTCR
36	IEANTRT
40	IEANTDL
44	DSPSERV EXTEND
48	ALESERV DELETE
Uppercase values are z/OS system macros. Lowercase values are internal requests.	

- zzzzzzzz = failing system service return code
- aaaaaaaa = failing system service reason code
- bbbbbbbb = identifies R_cacheserv subroutine
- cccccccc = additional diagnostic text, such as "dspCreate failed"

R_cacheserv (CSECTs IRRRCHQx in IRRRCH16)

- Error occurred in data space layer
  - Service name: DSPLAYER
  - Variable data: bbbbbbbb: cccccccc

- bbbbbbbb = identifies R_cacheserv subroutine
- cccccccc = additional diagnostic text

R_cacheserv (CSECTs IRRRCHQC in IRRRCH16)

- Recovery attempted after a failure during cache creation
  - Service name: REEXPOSE
  - Variable data: bbbbbbbb: RE–EXPOSE cccccccc dddddddd
    - bbbbbbbb = identifies R_cacheserv subroutine
    - cccccccc = identifies failing object
    - dddddddd = successful or failed

R_cacheserv (CSECT IRRRCHQD, IRRRCHQG, IRRRCHQE in IRRRCH16)

- Name/token service failure
  - Service name: IEANTRT
  - Variable data: bbbbbbbb: IEANTRT error
    - bbbbbbbb = identifies R_cacheserv subroutine

R_cacheserv (CSECT IRRRCHQG in IRRRCH16)

- Name/token service failure
  - Service name: IEANTCR
  - Variable data: bbbbbbbb: IEANTCR Error
    - bbbbbbbb = identifies R_cacheserv subroutine

R_cacheserv (CSECT IRRRCHQJ in IRRRCH16)

- Error restoring cache
  - Service name: RESTORE
  - Variable data: Restore: IRRRCH12 Restore DS error

R_cacheserv (CSECT IRRRCH18)

- Unable to use ICSF for random number generation, therefore, a software pseudo random number generator (SW PRNG) is used. It is possible that CSNBRNG is not found in LPA. See 'FINDMOD:' logrec for more information.
  - Service name: RANDNUM
  - Variable Data: IRRRCH18: USING SW PRNG, NOT CSNBRNG
  - Primary symptom string return code: 4
  - Primary symptom string reason code: 0
- Unable to find module in LPA using CSVQUERY
  - Service name: name of the module that was not found
  - Variable Data: *aaaaaaaa bbbbbbbb cccccccc dddddddd* FINDMOD: UNABLE TO FIND MODULE IN LPA.
    - *aaaaaaaa* is an indicator of R_cacheserv operation.

      The following are possible values for *aaaaaaaa*:

Table 42. Indicator of R_cacheserv operation

aaaaaaaa	**Operation**
201	IRRRCH10 SRB failed

Table 42. Indicator of R_cacheserv operation (continued)	
aaaaaaaa	**Operation**
202	Find IRRRCH10 SRB in LPA
203	Find other module in LPA (CSNBRNG)
210	Get cell from cell pool in data space
211	Add cell pool extent to cell pool in data space
213	Copy data to data cell in cell pool
214	Locate data in data space
215	Delete data in data space
216	R18ADD
217	R18IDX
218	R18GET
219	GenerateReference
220	CallDes

- *bbbbbbbb* is the system service or internal check being performed.

  The following are possible values for *bbbbbbbb*:

Table 43. System service or internal check being performed	
bbbbbbbb	**Service**
8	DSPSERV create – Create data space
10	CSRPGT1 – Get cell pool cell
11	CSRPEXP – Add cell pool extent
12	ALESERV Add
13	CHKEXT – Check if data space is full (does EXTent area hit data area)
14	Search – Find node in cache
15	PLO instruction #1
16	DSPSERV delete – delete data space
17	PLO instruction #2
18	CSRPFR1 – Free cell pool cell (ddddddddd = address of cell being freed)
19	Not used
20	Expiration time check for data or reference
21	CSNBRNG – ICSF random number function
22	Not used
23	CSVQUERY – Find mod in LPA
24	RACROUTE TYPE=ENCRYPT

- *cccccccc* is return code from service *bbbbbbbb*.

    - *dddddddd* is reason code from service *bbbbbbbb*.
  - Primary symptom string return code: 4
  - Primary symptom string reason code: 0
- CSNBRNG failure, using software pseudo random number generator (PRNG)
  - Service name: RANDNUM
  - Variable Data: *aaaaaaaa bbbbbbbb cccccccc dddddddd* GENREF: USE SOFTWARE PRNG, NOT CSNBRNG
    - *aaaaaaaa* = indicator of R_cacheserv operation

      See Table 42 on page 199 for possible values.
    - *bbbbbbbb* = system service or internal check being performed

      See Table 43 on page 200 for possible values.
    - *cccccccc* = return code from service *bbbbbbbb*
    - *dddddddd* = reason code from service *bbbbbbbb*
  - Primary symptom string return code: 0
  - Primary symptom string reason code: 0
- RACROUTE TYPE=ENCRYPT *failure*
  - Service name: CALLDES
  - Variable Data: *aaaaaaaa bbbbbbbb cccccccc dddddddd* RACROUTE REQ=EXTRACT,TYPE=ENCRYPT FAIL
    - *aaaaaaaa* = indicator of R_cacheserv operation

      See Table 42 on page 199 for possible values.
    - *bbbbbbbb* = system service or internal check being performed

      See Table 43 on page 200 for possible values.
    - *cccccccc* = return code from service *bbbbbbbb*
    - *dddddddd* = reason code from service *bbbbbbbb*
  - Primary symptom string return code: RACROUTE return code
  - Primary symptom string reason code: 0

R_cacheserv (CSECT IRRRCH20)

- Name/token service failure
  - Service name: [IEANTRT0 | IEANTRT1 | IEANTCR]
  - Variable data: SETUPCACHE: [IEANTRT|IEANTCR] FAILURE
  - Primary symptom string return code: return code from IEANTRT/IEANTCR
  - Primary symptom string reason code: 0 (name/token functions have no reason codes)
- Failure due to no RCVI
  - Service name: FINDRCVI
  - Variable data: RCVTRCVI=0. UNABLE TO BUILD ICTX
  - Primary symptom string return code: 0
  - Primary symptom string reason code: 0
- Error occurred during STORE operation
  - Service name: STORE
  - Variable Data: *aaaaaaaa bbbbbbbb cccccccc dddddddd* STORE: R18ADD ([1|2]) ERROR
    - *aaaaaaaa* = indicator of R_cacheserv operation

See Table 42 on page 199 for possible values.
- *bbbbbbbb* = system service or internal check being performed

See Table 43 on page 200 for possible values.
- *cccccccc* = return code from service *bbbbbbbb*
- *dddddddd* = reason code from service *bbbbbbbb*

– Primary symptom string return code: [0,4,8...]
– Primary symptom string reason code: 0

- Error occurred while an R_cacheserv operation was trying to find or create the cache
  – Service name: IRRRCH18
  – Variable Data: *aaaaaaaa bbbbbbbb cccccccc dddddddd* SETUPCACHE: ERROR IN IRRRCH18
    - *aaaaaaaa* = indicator of R_cacheserv operation

      See Table 42 on page 199 for possible values.
    - *bbbbbbbb* = system service or internal check being performed

      See Table 43 on page 200 for possible values.
    - *cccccccc* = return code from service *bbbbbbbb*
    - *dddddddd* = reason code from service *bbbbbbbb*
  – Primary symptom string return code: [0,4,8...]
  – Primary symptom string reason code: 0
- Error occurred during DESTROY operation
  – Service name: DESTROY
  – Variable Data: *aaaaaaaa bbbbbbbb cccccccc dddddddd* DESTROY: R18TERM ERROR
    - *aaaaaaaa* = indicator of R_cacheserv operation

      See Table 42 on page 199 for possible values.
    - *bbbbbbbb* = system service or internal check being performed

      See Table 43 on page 200 for possible values.
    - *cccccccc* = return code from service *bbbbbbbb*
    - *dddddddd* = reason code from service *bbbbbbbb*
  – Primary symptom string return code: [0,4,8...]
  – Primary symptom string reason code: 0
- Error occurred during RETRIEVE operation
  – Service name: RETRIEVE
  – Variable Data: *aaaaaaaa bbbbbbbb cccccccc dddddddd* RETRIEVE: R18GET ERROR
    - *aaaaaaaa* = indicator of R_cacheserv operation

      See Table 42 on page 199 for possible values.
    - *bbbbbbbb* = system service or internal check being performed

      See Table 43 on page 200 for possible values.
    - *cccccccc* = return code from service *bbbbbbbb*
    - *dddddddd* = reason code from service *bbbbbbbb*
  – Primary symptom string return code: [0,4,8...]
  – Primary symptom string reason code: 0
- Error occurred during REMOVE operation
  – Service name: REMOVE

  - Variable Data: *aaaaaaaa bbbbbbbb cccccccc dddddddd* STORE: R18DEL ERROR
    - *aaaaaaaa* = indicator of R_cacheserv operation

      See Table 42 on page 199 for possible values.
    - *bbbbbbbb* = system service or internal check being performed

      See Table 43 on page 200 for possible values.
    - *cccccccc* = return code from service *bbbbbbbb*
    - *dddddddd* = reason code from service *bbbbbbbb*
  - Primary symptom string return code: [0,4,8...]
  - Primary symptom string reason code: 0
- Error occurred during REMOVE EXPIRED operation
  - Service name: CLEANUP
  - Variable Data: *aaaaaaaa bbbbbbbb cccccccc dddddddd* STORE: R18CLN ERROR
    - *aaaaaaaa* = indicator of R_cacheserv operation

      See Table 42 on page 199 for possible values.
    - *bbbbbbbb* = system service or internal check being performed

      See Table 43 on page 200 for possible values.
    - *cccccccc* = return code from service *bbbbbbbb*
    - *dddddddd* = reason code from service *bbbbbbbb*
  - Primary symptom string return code: [0,4,8...]
  - Primary symptom string reason code: 0
- Error occurred during LOCATE operation
  - Service name: LOCATE
  - Variable Data: *aaaaaaaa bbbbbbbb cccccccc dddddddd* STORE: R18IDX ERROR
    - *aaaaaaaa* = indicator of R_cacheserv operation

      See Table 42 on page 199 for possible values.
    - *bbbbbbbb* = system service or internal check being performed

      See Table 43 on page 200 for possible values.
    - *cccccccc* = return code from service *bbbbbbbb*
    - *dddddddd* = reason code from service *bbbbbbbb*
  - Primary symptom string return code: [0,4,8...]
  - Primary symptom string reason code: 0

R_cacheserv (CSECT IRRRCH22)

- Error occurred obtaining storage for a DSG block
  - Service name: OBTAIN1
  - Variable data: *aaaaaaaa bbbbbbbb cccccccc dddddddd* STORAGE OBTAIN FOR DSG FAILURE
    - *aaaaaaaa* = STORAGE OBTAIN length
    - *bbbbbbbb* = STORAGE OBTAIN subpool
    - *cccccccc* = unused
    - *dddddddd* = unused
  - Primary symptom string return code: storage obtain return code. See *z/OS MVS Programming: Assembler Services Reference IAR-XCT* for the description of the return codes for the STORAGE OBTAIN macro.

  - Primary symptom string reason code: 0
- No active TTAB was found
  - Service name: TTAB1
  - Variable data: *aaaaaaaa bbbbbbbb cccccccc dddddddd* NO ACTIVE TTAB FOUND
    - *aaaaaaaa* = address of the DSCA
    - *bbbbbbbb* = TTAB anchor
    - *cccccccc* = active TTAB pointer
    - *dddddddd* = unused
  - Primary symptom string return code: 0
  - Primary symptom string reason code: 0
- Inconsistency found in TTAB slot
  - Service name: MEMTOKCK
  - Variable data: *aaaaaaaa bbbbbbbb cccccccc dddddddd* TTAB SLOT INCONSISTENCY system-name
    - *aaaaaaaa* = TTAB anchor
    - *bbbbbbbb* = active TTAB pointer
    - *cccccccc* = index into the active TTAB
    - *dddddddd* = system token for system name
    - system-name = system name of sysplex member
  - Primary symptom string return code: 0
  - Primary symptom string reason code: 0
- DSG is returned with either no DSGR or more than one DSGR
  - Service name: DSG2
  - Variable data: *aaaaaaaa bbbbbbbb cccccccc dddddddd* DSG INCONSISTENCY FROM SYSTEM system-name
    - *aaaaaaaa* = DSG address
    - *bbbbbbbb* = DSG return code
    - *cccccccc* = DSGR pointer
    - *dddddddd* = number of responses
    - system-name = system name of sysplex member
  - Primary symptom string return code: 0
  - Primary symptom string reason code: 0
- Error occurred when attempting intersystem communication
  - Service name: DSG1
  - Variable data: *aaaaaaaa bbbbbbbb cccccccc dddddddd* UNEXPECTED DSG RC FROM SYSTEM system-name
    - *aaaaaaaa* = DSG address
    - *bbbbbbbb* = DSG return code
    - *cccccccc* = number of DSGs sent
    - *dddddddd* = DSGR pointer
    - system-name = system name of sysplex member
  - Primary symptom string return code: 0
  - Primary symptom string reason code: 0

- Other system had either: more information to return and had a STORAGE OBTAIN error, or an unexpected error
  - Service name: IRRRCH24
  - Variable data: *aaaaaaaa bbbbbbbb ccccccc ddddddd* IRRRCH24 HAD PROBLEM ON SYSTEM system-name
    - *aaaaaaaa* = DSG address
    - *bbbbbbbb* = DSGR pointer
    - *ccccccc* = DSGR return buffer pointer
    - *ddddddd* = DSGR return buffer length
    - system-name = system name of sysplex member
  - Primary symptom string return code: IRRRCH24 return code
    - 2 - BAD_OBTAIN, STORAGE OBTAIN failure
  - Primary symptom string reason code: IRRRCH24 reason code
    - When IRRRCH24 return code = 2, storage obtain return code. See *z/OS MVS Programming: Assembler Services Reference IAR-XCT* for the description of the return codes for the STORAGE OBTAIN macro.
- The DSGR lacks a return buffer
  - Service name: DSGR1
  - Variable data: *aaaaaaaa bbbbbbbb ccccccc ddddddd* DSGR LACKS RETURN BUFFER FROM system-name
    - *aaaaaaaa* = DSG address
    - *bbbbbbbb* = DSGR pointer
    - *ccccccc* = DSGR return buffer pointer
    - *ddddddd* = DSGR return buffer length
    - system-name = system name of sysplex member
  - Primary symptom string return code: 0
  - Primary symptom string reason code: 0
- STORAGE OBTAIN failure on R_cacheserv return buffer for record name
  - Service name: OBTAIN2
  - Variable data: *aaaaaaaa bbbbbbbb ccccccc ddddddd* STORAGE OBTAIN FOR RECORD NAME FAILURE
    - *aaaaaaaa* = STORAGE OBTAIN length
    - *bbbbbbbb* = STORAGE OBTAIN subpool
    - *ccccccc* = unused
    - *ddddddd* = unused
  - Primary symptom string return code: storage obtain return code. See *z/OS MVS Programming: Assembler Services Reference IAR-XCT* for the description of the return codes for the STORAGE OBTAIN macro.
  - Primary symptom string reason code: 0
- STORAGE OBTAIN failure on R_cacheserv return buffer for date
  - Service name: OBTAIN3
  - Variable data: *aaaaaaaa bbbbbbbb ccccccc ddddddd* STORAGE OBTAIN FOR DATA FAILURE
    - *aaaaaaaa* = STORAGE OBTAIN length
    - *bbbbbbbb* = STORAGE OBTAIN subpool

    - *cccccccc* = unused
    - *dddddddd* = unused
  - Primary symptom string return code: storage obtain return code. See *z/OS MVS Programming: Assembler Services Reference IAR-XCT* for the description of the return codes for the STORAGE OBTAIN macro.
  - Primary symptom string reason code: 0
- STORAGE OBTAIN failure on R_cacheserv return buffer for source
  - Service name: OBTAIN4
  - Variable data: *aaaaaaaa bbbbbbbb cccccccc dddddddd* STORAGE OBTAIN FOR SOURCE FAILURE
    - *aaaaaaaa* = STORAGE OBTAIN length
    - *bbbbbbbb* = STORAGE OBTAIN subpool
    - *cccccccc* = unused
    - *dddddddd* = unused
  - Primary symptom string return code: storage obtain return code. See *z/OS MVS Programming: Assembler Services Reference IAR-XCT* for the description of the return codes for the STORAGE OBTAIN macro.
  - Primary symptom string reason code: 0
- Error from a R_cacheserv operation on another system
  - Service name: IRRRCH24
  - Variable data: *aaaaaaaa bbbbbbbb cccccccc dddddddd **xx* password ERROR ON SYSTEM system-name
    - *aaaaaaaa* = R_cacheserv option
    - *bbbbbbbb* = unused
    - *cccccccc* = unused
    - *dddddddd* = unused
    - ***xx* = 1st 4 characters of input user ID
    - password = input password
    - system-name = system name of sysplex member
  - Primary symptom string return code: system-name R_cacheserv RACF return code
  - Primary symptom string reason code: system-name R_cacheserv RACF reason code
- IRRRCH22 had an unknown return code
  - Service name: WAYOUT
  - Variable data: *aaaaaaaa bbbbbbbb cccccccc dddddddd **xx* password INTERNAL ERROR
    - *aaaaaaaa* = R_cacheserv option
    - *bbbbbbbb* = unused
    - *cccccccc* = unused
    - *dddddddd* = unused
    - ***xx* = 1st 4 characters of input user ID
    - password = input password
  - Primary symptom string return code: unexpected IRRRCH22 return code
  - Primary symptom string reason code: unexpected IRRRCH22 reason code

R_cacheserv (CSECT IRRRCH28)

- Name/token service failure

  - Service name: IEANTCR
  - Variable data: Identifies R_cacheserv subroutine
- ENVR object creation failure
  - Service name: IRRACM16
  - Variable data: Identifies subroutine and ACEE
- Unexpected return code from IRRRCH18 caching function
  - Service name: type of request (locate, store, and so on)
  - Variable data: Identifies related error information

R_cacheserv (CSECT IRRRCH30)

- Storage Obtain failure
  - Service name: OBTAINx
  - Variable data: Identifies subpool and length
- Unexpected return code from RACF sysplex communication path
  - Service name: Identifies the failure point
  - Variable data: Identifies related error information
- Unexpected return code from off-system cache
  - Service name: IRRRCH32
  - Variable data: system name and ICR information

R_Datalib (CSECT IRRRDL04)

- Error determining the owning user ID for a DIGTCERT profile
  - Service name: RACXTRT
  - Variable data: The string 'Method GETUSERID failed to extract the UserID'
  - Additional variable data: The profile name
- Error incrementing the last serial number (CERTLSER) of a DIGTCERT profile
  - Service name: RACXTRT or ICHEINTY
  - Variable data: The string 'Module IRRCDC29 failed to generate new Serial Number'
  - Additional variable data: The profile name

R_Datalib (CSECT IRRRDL06)

- Error parsing the input certificate
  - Service name: IRRCDC03
  - Variable data: The string ‘IRRCDC03 failed to parse the certificate’
- Error retrieving the certificate profile
  - Service name: RACMAP
  - Variable data: The string ‘RACMAP failed to retrieve the certificate profile’
  - Additional variable data: The certificate profile name
- Error retrieving the CERTNAME repeat group from a user profile
  - Service name: RACEXTR
  - Variable data: The string ‘RACEXTR failed to retrieve the CERTNAME repeat group’
  - Additional variable data: The user name
- Error locating the certificate in the CERTNAME repeat group
  - Service name: RACEXTR

  - Variable data: The String ‘Certificate not found in USER profile’
  - Additional variable data: The user name
- Error retrieving the private key from the certificate profile
  - Service name: RACEXTR
  - Variable data: The string ‘RACFEXTR failed to retrieve the private key’
  - Additional variable data: The certificate profile name
- Error adding the certificate to RACF
  - Service name: IRRCDC02
  - Variable data: The string 'IRRCDC02 Error - Certificate already exists'
  - Additional variable data: The certificate profile name
- Error adding the certificate to RACF
  - Service name: IRRCDC02
  - Variable data: The string ‘IRRCDC02 failed to add the certificate’
  - Additional variable data: The certificate profile name
- Error altering the certificate in RACF
  - Service name: IRRCDC05
  - Variable data: The string ‘IRRCDC05 failed to alter the certificate’
  - Additional variable data: The certificate profile name
- Error connecting the certificate to a RACF ring
  - Service name: IRRCDC19
  - Variable data: The string ‘IRRCDC19 failed to connect the certificate’
  - Additional variable data: The certificate profile name
- Error calling ICSF service CSNDKRR to retrieve the PKDS entry
  - Service name: CSNDKRR
  - Variable data: The string ‘CSNDKRR failed to retrieve the PKDS entry’
  - Additional variable data: The PKDS entry name

R_Datalib (CSECT IRRRDL10)

- Error altering the certificate in RACF
  - Service name: IRRCDC05
  - Variable data: The string 'IRRCDC05 failed to alter the certificate'
  - Additional variable data: The certificate label name

R_Datalib (CSECT IRRRDL12)

- Error verifying that the specified ring profile exists in RACF
  - Service name: RACEXTR
  - Variable data: The string 'RACEXTR failed to retrieve the DIGTRING profile'
  - Additional variable data: The ring profile name
- Error verifying that the specified ring owner exists in RACF
  - Service name: RACEXTR
  - Variable data: The string 'RACEXTR failed to retrieve the USER profile'
  - Additional variable data: The user name
- Error retrieving the next DIGTRING profile from RACF
  - Service name: RACEXTRN

  - Variable data: The string 'RACEXTRN failed to retrieve the next DIGTRING profile'
  - Additional variable data: The ring profile name
- Error retrieving the CERTDATA segment CERTNAME and CERTLABL repeat group fields from RACF
  - Service name: RACEXTR
  - Variable data: The string 'RACEXTR failed to retrieve ring information'
  - Additional variable data: The certificate profile name
- Error retrieving the certificate owner from RACF
  - Service name: RACMAP
  - Variable data: The string 'RACMAP failed to retrieve the certificate owner'
  - Additional variable data: The certificate profile name

R_DCEAUTH (CSECT IRRRDA00)

- Create ACEE failure
  - Service name: RACINIT
  - Variable data: the RACF user ID
- Convert UUID to RACF user ID failure
  - Service name: RACXTRT
  - Variable data: the DCEUUIDS profile name
- Check authorization failure
  - Service name: RACHECK or FASTAUTH
  - Variable data: the RACF user ID, class, profile, and access requested
- Delete ACEE failure
  - Service name: RACINIT
  - Variable data: the ACEE address

R_DCEINFO (CSECT IRRRDI00)

- Extract DCE fields failure
  - Service name: RACXTRT
  - Variable data: the RACF user ID and the names of the fields
- Replace DCE fields failure
  - Service name: RACXTRT
  - Variable data: the RACF user ID and the names of the fields

R_DCERUID (CSECT IRRRUD00)

- Extract DCEUUIDS APPLDATA failure
  - Service name: RACXTRT
  - Variable data: the DCEUUIDS profile name

R_DCEKEY (CSECT IRRRDK00)

- Extract DCE key failure
  - Service name: RACXTRT
  - Variable data: the RACF user ID
- Replace DCE key failure
  - Service name: RACXTRT
  - Variable data: the RACF user ID

R_DCEKEY (CSECT IRRCPT01)

- Extract KEYSMSTR key failure
  - Service name: ICHEINTY
  - Variable data: the class and profile name
- ICSF CSNBENC callable service failure
  - Service name: CSNBENC
  - Variable data: the RACF user ID
- ICSF CSNBDEC callable service failure
  - Service name: CSNBDEC
  - Variable data: the RACF user ID

R_PKIserv (CSECT IRRRPX00)

- Error calling RACROUTE REQUEST=AUTH to determine the caller's access to a FACILITY class profile
  - Service name: RACHECK
  - Variable data: The string 'Error in R_PKIserv attempting to check a caller's authority to the service'
  - Additional variable data: The FACILITY class profile name, the ACEE address, and the user ID from the ACEE

R_PKIserv (CSECT IRRRPX02)

- Error calling RACROUTE REQUEST=AUTH to determine the caller's access to a FACILITY class profile
  - Service name: RACHECK
  - Variable data: The string 'Error in R_PKIserv GENCERT attempting to check a caller's authority to the service'
  - Additional variable data: The FACILITY class profile name, the ACEE address, and the user ID from the ACEE
- Unexpected return code from RACDCERT internal subroutine IRRCDC23
  - Service name: IRRCDC23
  - Variable data: The string 'Error in R_PKIserv GENCERT attempting to call RACDCERT subroutine. Unexpected return code'
  - Additional variable data: The first 942 bytes of the RACDCERT internal work area
- Error calling RACROUTE REQUEST=EXTRACT to retrieve the PGMRNAME field from the USER profile
  - Service name: RACXTRT
  - Variable data: The string 'Error in R_PKIserv GENCERT attempting to extract PGMRNAME field to build Common Name'
  - Additional variable data: The user ID

R_PKIserv (CSECT IRRRPX04)

- Error calling RACROUTE REQUEST=AUTH to determine the caller's access to a FACILITY class profile
  - Service name: RACHECK
  - Variable data: The string 'Error in R_PKIserv EXPORT attempting to check the caller's authority to the service'
  - Additional variable data: The FACILITY class profile name, the ACEE address, and the user ID from the ACEE

- Unexpected return code from RACDCERT internal subroutine IRRCDC25
  - Service name: IRRCDC25
  - Variable data: The string 'Error in R_PKIserv EXPORT attempting to call RACDCERT subroutine. Unexpected return code'
  - Additional variable data: The first 942 bytes of the RACDCERT internal work area

R_PKIserv (CSECT IRRRPX06)

- Error calling RACROUTE REQUEST=AUTH to determine the caller's access to a FACILITY class profile
  - Service name: RACHECK
  - Variable data: The string 'Error in R_PKIserv GENCERT attempting to check the caller's authority to the service'
  - Additional variable data: The FACILITY class profile name, the ACEE address, and the user ID from the ACEE
- Unexpected return code from RACDCERT internal subroutine IRRCDC27
  - Service name: IRRCDC27
  - Variable data: The string 'Error in R_PKIserv GENCERT attempting to call RACDCERT subroutine. Unexpected return code'
  - Additional variable data: The first 942 bytes of the RACDCERT internal work area
- Error calling RACROUTE REQUEST=EXTRACT to retrieve the PGMRNAME field from the USER profile
  - Service name: RACXTRT
  - Variable data: The string 'Error in R_PKIserv GENCERT attempting to extract PGMRNAME field to build Common Name'
  - Additional variable data: The user ID
- Error calling MVS Name/Token Service
  - Service name: IEANTRT
  - Variable data: The string 'Name/Token service failed'
  - Additional variable data: The first 942 bytes of the RACDCERT internal work area
- Error on PC to PKI Services
  - Service name: PKIGENC
  - Variable data: The string 'PKI Services Failed for function GENCERT'
  - Additional variable data: Additional information word returned from PKI Services (register 1) followed by the parameter list used on the PC

R_PKIserv (CSECT IRRRPX07)

- Error calling RACROUTE REQUEST=AUTH to determine the caller's access to a FACILITY class profile
  - Service name: RACHECK
  - Variable data: The string 'Error in R_PKIserv EXPORT attempting to check the caller's authority to the service'
  - Additional variable data: The FACILITY class profile name, the ACEE address, and the user ID from the ACEE
- Error calling IRRCDC67 to find parent CERTAUTH chain to input certificate and build PKCS7 package.
  - Service name: IRRCDC67
  - Variable data: The string 'Error in R_PKIServ EXPORT attempting to call RACDCERT subroutine. Unexpected return code.'

  - Additional variable data: RACDCERT failing module and service in form of diagnostic data from the RACDCERT internal work area:
    - 4 bytes - internal service code
    - 4 bytes - service return code
    - 4 bytes - service reason code
    - 8 bytes - RACF class name
    - 8 bytes - RACF segment name
    - 81 bytes - blank
    - 8 bytes - csect name
    - 71 bytes - diagnostic message text
    - 4 bytes - RACF entity length
    - 4 bytes - number of extracted fields
    - 56 bytes - array of 8 byte field names (max 7 entries)
    - 256 bytes - RACF entity name
- Error calling MVS Name/Token Service
  - Service name: IEANTRT
  - Variable data: The string 'Name/Token service failed'
  - Additional variable data: None
- Error on PC to PKI Services
  - Service name: PKIEXPT
  - Variable data: The string 'PKI Services Failed for function EXPORT'
  - Additional variable data: Additional information word returned from PKI Services (register 1) followed by the parameter list used on the PC

R_PKIserv (CSECT IRRRPX08)

- Error calling MVS Name/Token Service
  - Service name: IEANTRT
  - Variable data: The string 'Name/Token service failed'
  - Additional variable data: None
- Error on PC to PKI Services
  - Service name: PKIQRYR or PKIQRYC
  - Variable data: The string 'PKI Services Failed for function {QUERYREQS | QUERYCERTS}'
  - Additional variable data: Additional information word returned from PKI Services (register 1) followed by the parameter list used on the PC

R_PKIserv (CSECT IRRRPX10)

- Error calling MVS Name/Token Service
  - Service name: IEANTRT
  - Variable data: The string 'Name/Token service failed'
  - Additional variable data: None
- Error on PC to PKI Services
  - Service name: PKICRTR or PKICRTD
  - Variable data: The string 'PKI Services Failed for function {REQDETAILS | CERTDETAILS}'

  - Additional variable data: Additional information word returned from PKI Services (register 1) followed by the parameter list used on the PC

R_PKIserv (CSECT IRRRPX12)

- Error on PC to PKI Services
  - Service name: PKIMODR or PKIMODC
  - Variable data: The string 'PKI Services Failed for function {MODIFYREQS | MODIFYCERTS}'
  - Additional variable data: Additional information word returned from PKI Services (register 1) followed by the parameter list used on the PC
- Error calling MVS Name/Token Service
  - Service name: IEANTRT
  - Variable data: The string 'Name/Token service failed'
  - Additional variable data: None

R_PKIserv (CSECT IRRRPX13)

- Error calling RACROUTE REQUEST=AUTH to determine the caller's access to a FACILITY class profile
  - Service name: RACHECK
  - Variable data: The string 'Error in R_PKIServ VERIFY attempting to check the caller's authority to the service'
  - Additional variable data: The FACILITY class profile name, the ACEE address, and the user ID from the ACEE
- Error calling MVS Name/Token Service
  - Service name: IEANTRT
  - Variable data: The string 'Name/Token service failed'
  - Additional variable data: None
- Error on PC to PKI Services
  - Service name: PKIVERF
  - Variable data: The string 'PKI Services Failed for function VERIFY'
  - Additional variable data: Additional information word returned from PKI Services (register 1) followed by the parameter list used on the PC

R_PKIserv (CSECT IRRRPX14)

- Error calling RACROUTE REQUEST=AUTH to determine the caller's access to a FACILITY class profile
  - Service name: RACHECK
  - Variable data: The string 'Error in R_PKIServ REVOKE attempting to check the caller's authority to the service'
  - Additional variable data: The FACILITY class profile name, the ACEE address, and the user ID from the ACEE
- Error calling MVS Name/Token Service
  - Service name: IEANTRT
  - Variable data: The string 'Name/Token service failed'
  - Additional variable data: None
- Error on PC to PKI Services
  - Service name: PKIREVOKE

  - Variable data: The string 'PKI Services Failed for function REVOKE'
  - Additional variable data: Additional information word returned from PKI Services (register 1) followed by the parameter list used on the PC

R_PKIserv (CSECT IRRRPX15)

- Error calling RACROUTE REQUEST=AUTH to determine the caller's access to a FACILITY class profile
  - Service name: RACHECK
  - Variable data: The string 'Error in R_PKIServ RENEW attempting to check the caller's authority to the service'
  - Additional variable data: The FACILITY class profile name, the ACEE address, and the user ID from the ACEE
- Error on PC to PKI Services
  - Service name: PKIRENW
  - Variable data: The string 'PKI Services Failed for function RENEW'
  - Additional variable data: Additional information word returned from PKI Services (register 1) followed by the parameter list used on the PC
- Error calling MVS Name/Token Service
  - Service name: IEANTRT
  - Variable data: The string 'Name/Token service failed'
  - Additional variable data: None

R_PKIserv (CSECT IRRRPX18)

- Error calling RACROUTE REQUEST=AUTH to determine the caller's access to a FACILITY class profile
  - Service name: RACHECK
  - Variable data: The string 'Error in R_PKIServ RESPOND attempting to check the caller's authority to the service'
  - The FACILITY class profile name, the ACEE address, the user ID from the ACEE
- Error calling MVS Name/Token Service
  - Service name: IEANTRT
  - Variable data: The string 'Name/Token Service, IEANTRT, failed'
  - Additional variable data: None
- Error on PC to PKI Services
  - Service name: R_PKIServ RESPOND
  - Variable data: The string 'PKI Services failed for function RESPOND'
  - Additional variable data: Additional information word returned from PKI Services (register 1) followed by the parameter list used on the PC.

R_PKIserv (CSECT IRRRPX20)

- Error calling RACROUTE REQUEST=AUTH to determine the caller's access to a FACILITY class profile
  - Service name: RACHECK
  - Variable data: The string 'Error in R_PKIServ SCEPREQ attempting to check the caller's authority to the service'
  - Additional variable data: The FACILITY class profile name, the ACEE address, the user ID from the ACEE

- Error calling MVS Name/Token Service
  - Service name: IEANTRT
  - Variable data: The string 'Name/Token Service, IEANTRT, failed'
  - Additional variable data: None
- Error on PC to PKI Services
  - Service name: R_PKIServ PKISCEP
  - Variable data: The string 'PKI Services failed for function SCEPREQ'
  - Additional variable data: Additional information word returned from PKI Services (register 1) followed by the parameter list used on the PC.

R_PKIserv (CSECT IRRRPX22)

- Error calling MVS Name/Token Service
  - Service name: IEANTRT
  - Variable data: The string 'Name/Token Service, IEANTRT, failed'
  - Additional variable data: None
- Error on PC to PKI Services
  - Service name: R_PKIServ PKIPREG
  - Variable data: The string 'PKI Services failed for function PREREGISTER'
  - Additional variable data: Additional information word returned from PKI Services (register 1) followed by the parameter list used on the PC.

R_PKIserv (CSECT IRRRPX24)

- Error calling RACROUTE REQUEST=AUTH to determine the caller's access to a FACILITY class profile
  - Service name: RACHECK
  - Variable data: The string 'Error in R_PKIserv QRECOVER attempting to check the caller's authority to the service'
  - Additional variable data: The FACILITY class profile name, the ACEE address, and the user ID from the ACEE
- Error calling MVS Name/Token Service
  - Service name: IEANTRT Error recording
  - Variable data: The string 'Name/Token service failed'
  - Additional variable data: None
- Error on PC to PKI Services
  - Service name: PKIQREC
  - Variable data: The string 'PKI Services Failed for function QRECOVER'
  - Additional variable data: Additional information word returned from PKI Services (register 1), followed by the parameter list used on the PC

R_PgmSignVer (CSECT IRRSPS00)

- Error with range
  - Service name: IRRRPS00
  - Variable data: Too many ranges
  - Primary symptom string return code: Index of invalid range
  - Secondary symptom string reason code: 0
- Error with end address

  - Service name: IRRRPS00
  - Variable data: End Address before start

    ```
    Start: 0Xaaaaaaaa aaaaaaaa
    End: 0Xbbbbbbbb bbbbbbbb"
    ```

    - Aaaaaaa aaaaaaaa=start address
    - Bbbbbbbb bbbbbbbb = end address
  - Primary symptom string return code: Index of invalid range
  - Secondary symptom string reason code: 0
- Extract error
  - Service name: SIGINIT
  - Variable data: Extract error:

    ```
    0xaaaaaaaa
    0xbbbbbbbb
    0xcccccccc
    ```

    - Aaaaaaaa=SAFRC
    - Bbbbbbb=RACFRC
    - Ccccccc=RACFRSN

    Primary symptom string return code: RACROUTE return code

    Secondary symptom string reason code: RACROUTE reason code
- Error with APPLDATA algorithm
  - Service name: SIGINIT
  - Variable data: APPLDATA invalid algorithm: `alg`
    - Alg=What was specified in appldata
  - Primary symptom string return code: 0
  - Secondary symptom string reason code: 0
- Error locating APPLDATA in profile
  - Service name: SIGINIT
  - Variable data: APPLDATA not found in profile
  - Primary symptom string return code: 0
  - Secondary symptom string reason code: 0
- APPLDATA error with leading/trailing blanks
  - Service name: SIGINIT
  - Variable data: APPLDATA: Leading/Trailing blank encountered
  - Primary symptom string return code: Variable 'I' – Index of first blank in appldata
  - Secondary symptom string reason code: 0
- Error with APPLDATA algorithm name
  - Service name: SIGINIT
  - Variable data: APPLDATA: algorithm name too long
  - Primary symptom string return code: Variable 'I' – Index of first blank in appldata
  - Secondary symptom string reason code: 0
- RING error, ID too long
  - Service name: SIGINIT

  - Variable data: RING: / not found or ID too long
  - Primary symptom string return code: Variable ‘I’, index of ‘/’
  - Secondary symptom string reason code: Variable ‘j’ Index of ‘ ‘
- RING error, name too long
  - Service name: SIGINIT
  - Variable data: RING: Name too long or not specified
  - Primary symptom string return code: Variable ‘I’, index of ‘/’
  - Secondary symptom string reason code: Variable ‘j’ Index of ‘ ‘
- Key error in default certificate
  - Service name: SIGINIT
  - Variable data: No Private Key in default certificate
  - Primary symptom string return code: Number of certificates in ring
  - Secondary symptom string reason code: Index of default certificate in ring
- Error with key analysis
  - Service name: SIGINIT
  - Variable data: Key analysis error
  - Primary symptom string return code: RC from analysisKey() routine
  - Secondary symptom string reason code: 0
- Error with number of certificates in ring
  - Service name: SIGINIT
  - Variable data: Too many certificates in ring
  - Primary symptom string return code: Number of certificates in ring
  - Secondary symptom string reason code: 0
- Error with ring default certificate
  - Service name: SIGINIT
  - Variable data: No default certificate found in ring
  - Primary symptom string return code: 0
  - Secondary symptom string reason code: 0
- Error with certificate ordering
  - Service name: SIGINIT
  - Variable data: Certificate ordering error
  - Primary symptom string return code: RC from OrderPKCS7CA
  - Secondary symptom string reason code: 0
- Error with trust chain length
  - Service name: SIGINIT
  - Variable data: Trust chain too long
  - Primary symptom string return code: Number of items in trust chain
  - Secondary symptom string reason code: 0
- Error with R_datalib
  - Service name: SIGINIT
  - Variable data: R_datalib error:

```
Function Code X,
RC=0Xaaaaaaaa 0Xbbbbbbbb 0Xcccccccc
```

  - Primary symptom string return code: R_datalib Function code
  - Secondary symptom string reason code: R_datalib SAFRC
- Digest failure error
  - Service name: IRRRPS21
  - Variable data: Digest failure
  - Primary symptom string return code: 0
  - Secondary symptom string reason code: 0
- Digest failure 2 error
  - Service name: IRRRPS21
  - Variable data: Digest failure 2
  - Primary symptom string return code: 0
  - Secondary symptom string reason code: 0
- Number of ranges error
  - Service name: IRRRPS21
  - Variable data: Too many ranges
  - Primary symptom string return code: # specified ranges
  - Secondary symptom string reason code: 0
- Error with end address
  - Service name: IRRRPS21
  - Variable data: End address before start:

    ```
    Start: 0Xaaaaaaaa aaaaaaaa
    End: 0Xbbbbbbbb bbbbbbbb"
    ```

    - Aaaaaaa aaaaaaaa=start address
    - Bbbbbbbb bbbbbbbb = end address
  - Primary symptom string return code: Failing range index
  - Secondary symptom string reason code: 0
- Context error
  - Service name: IRRRPS21
  - Variable data: New context
  - Primary symptom string return code: CLiC rc
  - Secondary symptom string reason code: 0
- New digest error
  - Service name: IRRRPS21
  - Variable data: New digest
  - Primary symptom string return code: CLiC rc
  - Secondary symptom string reason code: 0
- Digest update error
  - Service name: IRRRPS21
  - Variable data: Digest update
  - Primary symptom string return code: CLiC rc

  - Secondary symptom string reason code: 0
- Digest update 2 error
  - Service name: IRRRPS21
  - Variable data: Digest update 2
  - Primary symptom string return code: CLiC rc
  - Secondary symptom string reason code: 0
- New context error
  - Service name: IRRRPS31
  - Variable data: New context
  - Primary symptom string return code: CLiC rc
  - Secondary symptom string reason code: 0
- Pk service error
  - Service name: IRRRPS31
  - Variable data: Pk service
  - Primary symptom string return code: CLiC rc
  - Secondary symptom string reason code: 0
- Rsa sign error
  - Service name: IRRRPS31
  - Variable data: Rsa sign
  - Primary symptom string return code: CLiC rc
  - Secondary symptom string reason code: 0
- Rsa verify error
  - Service name: IRRRPS31
  - Variable data: Rsa verify
  - Primary symptom string return code: CLiC rc
  - Secondary symptom string reason code: 0
- Decode failure 1 error
  - Service name: IRRRPS51
  - Variable data: Decode Failure 1
  - Primary symptom string return code: RC from decodeSimple()
  - Secondary symptom string reason code: 0
- Decode failure 2 error
  - Service name: IRRRPS51
  - Variable data: Decode Failure 2

    ```
    aaaaaaaa bbbbbbbb cccccccc dddddddd
    eeeeeeee ffffffff gggggggg hhhhhhhh iiiiiiiiI
    ```

  - Primary symptom string return code: 0
  - Secondary symptom string reason code: 0
- Decode failure 3 error
  - Service name: IRRRPS51
  - Variable data: Decode Failure 3 (hex data)
  - Primary symptom string return code: 0

  - Secondary symptom string reason code: 0
- Decode failure 4 error
  - Service name: IRRRPS51
  - Variable data: Decode failure 4
  - Primary symptom string return code: RC from DecodeSimple
  - Secondary symptom string reason code: 0
- Decode failure 5 error
  - Service name: IRRRPS51
  - Variable data: Decode failure 5
  - Primary symptom string return code: RC from analysiskey()
  - Secondary symptom string reason code: 0
- Decode failure 6 error
  - Service name: IRRRPS51
  - Variable data: Decode failure 6
  - Primary symptom string return code: Value of variable seqCount should be 5
  - Secondary symptom string reason code: 0
- Error with signature algorithm
  - Service name: VERFINAL
  - Variable data: Unsupported signature algorithm
  - Primary symptom string return code: Value of CxSignAlg
  - Secondary symptom string reason code: 0
- APPLDATA format error
  - Service name: VERFINAL
  - Variable data: Appldata format error:

    ```
    #### aaaaa
    ```

  - Primary symptom string return code: Variable BlankPos: Index of ‘ ‘ in Appldata
  - Secondary symptom string reason code: Variable DelimPos: index of ‘/’ in ApplData
- Extract error
  - Service name: VERFINAL
  - Variable data: Extract Error:

    ```
    0xaaaaaaaa
    0xbbbbbbbb
    0xcccccccc
    ```

    - Aaaaaaa=safrc
    - Bbbbbbbb=racfrc
    - Cccccccc=racfrsn
  - Primary symptom string return code: RACROUTE
  - Secondary symptom string reason code: RACROUTE
- Decode X509 error
  - Service name: VERFINAL
  - Variable data: Decode X509 error
  - Primary symptom string return code: RC from Decode X509

  - Secondary symptom string reason code: Certificate number (I loop index)
- SetCertificate error
  - Service name: VERFINAL
  - Variable data: SetCertificate Error
  - Primary symptom string return code: RC from SetCertificate
  - Secondary symptom string reason code: Certificate number
- R_datalib error
  - Service name: VERFINAL
  - Variable data: R_datalib error:

```
Function code aaaaaaaa,
RC=0xbbbbbbbb 0xcccccccc 0xddddddddd
```

    - Aaaaaaaa=r_datalib func code
    - Bbbbbbbb=saf RC from r_datalib
    - Cccccccc=RACFRC from r_datalib
    - Ddddddddd=RACFRsn from r_datalib
  - Primary symptom string return code: R_datalib function code
  - Secondary symptom string reason code: R_datalib RC

R_proxyserv (CSECT IRRRPY00)

- Name/token service failure
  - Service name: IEANTRT
  - Variable data: The string "Name/token service failure"
  - Additional variable data: The name for the requested token

R_proxyserv (CSECT IRRRPY01 and IRRRPY02)

- Error response from LDAP
  - Service name: LDAP
  - Variable data: Up to 100 bytes of the LDAP error message
- LDAP message response decoding failure
  - Service name: LDAPMSG
  - Variable data: One of the following strings –
    - "Error in LDAP message – Incorrect ASN.1 syntax"
    - "Error in LDAP message – Expected field not found"
    - "Error in LDAP message – Unsupported character set detected"
  - Additional variable data: Up to 1024 bytes of the returned LDAP message

R_proxyserv (CSECT IRRCPT01)

- Extract KEYSMSTR key failure
  - Service name: ICHEINTY
  - Variable data: the class and profile name
- ICSF CSNBENC callable service failure
  - Service name: CSNBENC
  - Variable data: the RACF user ID
- ICSF CSNBDEC callable service failure

- Service name: CSNBDEC
- Variable data: the RACF user ID

R_admin (CSECT IRRREQ00)

- The R_admin callable service uses the RACF subsystem address space. RACF command images generated by the callable service or by the caller are passed to the RACF subsystem for execution by invoking the internal macro IRRREQ. The IRRREQ macro uses the IEFSSREQ macro to communicate with the RACF subsystem.

  In this environment the following errors could occur:

  - IRRREQ is unable to communicate with the RACF subsystem (for example, the RACF subsystem is not started).
  - IRRREQ is able to communicate with the RACF subsystem, but the RACF command fails in the RACF subsystem (for example, an ADDUSER command specifying a class that is not valid).
- The R_admin callable service differentiates between these two errors. For the first error, the callable service returns a SAF return code of 8, a RACF return code of 12, and a RACF reason code set equal to the return code from IEFSSREQ. For the second error, the callable service returns a SAF return code of 8, a RACF return code of 16, and a RACF reason code set equal to the return code returned from the command's execution in the RACF subsystem. Additionally, the R_admin callable service writes symptom records to SYS1.LOGREC for the errors. The symptom records contain the following information:

  - Primary symptom string:
    - Component identifier (PIDS) - 5752XXH00
    - 1st return code (PRCS) - RACF return code from the callable service
    - 2nd return code (PRCS) - RACF reason code from the callable service
    - Csect routine identifier (RIDS) - IRRREQ00
    - Loadmod routine identifier (RIDS) - IRRRFR10#L
  - Secondary symptom string:
    - Storage address (ADRS) - Error offset into IRRREQ00
  - Free format component information: Specific to the error

  For the first error (unable to communicate with the RACF subsystem), RACF records the following additional information in the free format area:

Table 44. Additional information

Offset	Length	Description
0	1	R_admin callable service function requested
1	4	Address of the command image to be executed
5	2	Length of the command image to be executed

  Figure 65 on page 223 shows an example of a symptom record created when the R_admin callable service is unable to communicate with the RACF subsystem.

```
TYPE: SYMPTOM RECORD  REPORT: SOFTWARE EDIT REPORT  DAY YEAR
                              REPORT DATE: 348  96
SCP:   VS 2 REL 3              ERROR DATE: 348  96
                     MODEL:  4381                        HH MM SS.TH
                     SERIAL: 111606                TIME: 15:24:51.20
SEARCH ARGUMENT ABSTRACT:
     PIDS/5752XXH00 PRCS/0000000C PRCS/00000004 RIDS/IRRREQ00
     RIDS/IRRRFR10#L

SYSTEM ENVIRONMENT:
     CPU MODEL:  4381              DATE:  348  96
     CPU SERIAL: 111606            TIME:  15:24:51.20
     SYSTEM:     NODE1             BCP:   MVS
     RELEASE LEVEL OF SERVICE ROUTINE:      JBB6604
     SYSTEM DATA AT ARCHITECTURE LEVEL:     10
     COMPONENT DATA AT ARCHITECTURE LEVEL:  10
     SYSTEM DATA:  00000000 00000000    |........|
COMPONENT INFORMATION:
     COMPONENT ID:             5752XXH00
     COMPONENT RELEASE LEVEL:  240
     SERVICE RELEASE LEVEL:    HRF2240
     DESCRIPTION OF FUNCTION:  RACF z/OS UNIX SYSTEM CALLABLE SERV.
     PROBLEM ID:               IRRREQ

PRIMARY SYMPTOM STRING:
     PIDS/5752XXH00 PRCS/0000000C PRCS/00000004 RIDS/IRRREQ00
     RIDS/IRRRFR10#L

     SYMPTOM           SYMPTOM DATA     EXPLANATION
    ---------------    -------------    -----------
    PIDS/5752XXH00     5752XXH00        COMPONENT IDENTIFIER
    PRCS/0000000C      0000000C         RETURN CODE
    PRCS/00000004      00000004         RETURN CODE
    RIDS/IRRREQ00      IRRREQ00         ROUTINE IDENTIFIER
    RIDS/IRRRFR10#L    IRRRFR10#L       ROUTINE IDENTIFIER

SECONDARY SYMPTOM STRING:
    ADRS/00000734

     SYMPTOM            SYMPTOM DATA     EXPLANATION
    ---------------    -------------    -----------
     ADRS/00000734      00000734         STORAGE ADDRESS

FREE FORMAT COMPONENT INFORMATION:
 KEY = F000     LENGTH = 000100 (0064)
+000   0204803C    A2000D40    40404040    40404040    |....S..         |
+010 LENGTH(0080) ==> ALL BYTES CONTAIN X'40'.
+060   40404040                                        |                |
```

Figure 65. An example of a symptom record when the callable service cannot communicate with RACF

The IRRREQ (IEFSSREQ) return code is 4. Looking at the free format component information, the first byte has a value of X'02', indicating this was an ADMN_DEL_USER request. The next 4 bytes contain the address of the generated command image (X'04803CA2'). The sixth and seventh bytes contain the two byte length of the generated command image (X'000D').

For the second error (execution of the RACF command failed), the main intent of the symptom record is to capture the command image that was executed, because it might not be available elsewhere. Therefore, to limit the amount of user error information written to SYS1.LOGREC, RACF does not write symptom records for ADMN_RUN_COMD requests that fail. RACF also does not write symptom records for ADMN_ADD_USER requests where the RACF user ID exists and ADMN_DEL_USER requests where the RACF user ID does not exist. For the cases where RACF does write a symptom record, the following additional information is recorded in the free format area:

- First X'64' byte area–Command message output (if any). This is the first message text string only and is truncated if it exceeds X'64' bytes in length.
- Next four X'100' byte areas–A copy of the command image that was executed. The command image is truncated if it exceeds 1024 (decimal) bytes in length.

You can trace the R_admin callable service to obtain information not included in the symptom records. For information about tracing, see “Tracing the R_admin callable service” on page 57.

Figure 66 on page 224 shows an example of a symptom record created when the R_admin callable service is able to communicate with the RACF subsystem, but the command fails.

```
TYPE: SYMPTOM RECORD      REPORT: SOFTWARE EDIT REPORT  DAY YEAR
                          REPORT DATE:     014  97
SCP:   VS 2 REL 3         ERROR DATE:      014  97
                          MODEL:  4381     HH   MM   SS.TH
                          SERIAL: 111606   TIME: 10:24:10.11
SEARCH ARGUMENT ABSTRACT:
    PIDS/5752XXH00 PRCS/00000010 PRCS/00000008 RIDS/IRRREQ00
    RIDS/IRRRFR10#L
    SYSTEM ENVIRONMENT:
    CPU MODEL:  4381              DATE:  014  97
    CPU SERIAL: 111606            TIME:  10:24:10.11
    SYSTEM:     NODE1             BCP:   MVS
    RELEASE LEVEL OF SERVICE ROUTINE:      JBB6604
    SYSTEM DATA AT ARCHITECTURE LEVEL:     10
    COMPONENT DATA AT ARCHITECTURE LEVEL:  10
    SYSTEM DATA:  00000000 00000000    |........|
COMPONENT INFORMATION:
    COMPONENT ID:             5752XXH00
    COMPONENT RELEASE LEVEL:  240
    SERVICE RELEASE LEVEL:    HRF2240
    DESCRIPTION OF FUNCTION:  RACF z/OS UNIX SYSTEM CALLABLE SERV.
    PROBLEM ID:               IRRREQ
    PRIMARY SYMPTOM STRING:
    PIDS/5752XXH00 PRCS/00000010 PRCS/00000008 RIDS/IRRREQ00
    RIDS/IRRRFR10#L
```

```
    SYMPTOM            SYMPTOM DATA     EXPLANATION
    ---------------    -------------    -----------
    PIDS/5752XXH00     5752XXH00        COMPONENT IDENTIFIER
    PRCS/00000010      00000010         RETURN CODE
    PRCS/00000008      00000008         RETURN CODE
    RIDS/IRRREQ00      IRRREQ00         ROUTINE IDENTIFIER
    RIDS/IRRRFR10#L    IRRRFR10#L       ROUTINE IDENTIFIER
    SECONDARY SYMPTOM STRING:
    ADRS/00000748

    SYMPTOM            SYMPTOM DATA     EXPLANATION
    ---------------    -------------    -----------

    ADRS/00000748      00000748         STORAGE ADDRESS
 FREE FORMAT COMPONENT INFORMATION:
 KEY = F000     LENGTH = 000100 (0064)
+000   C9D2D1F5    F6F7F0F2    C940C9D5    E5C1D3C9    |IKJ56702I INVALI|
+010   C440C3D3    C1E2E240    D5C1D4C5    6B40C2C1    |D CLASS NAME, BA|
+020   C4C3D3C1    E2E24040    40404040    40404040    |DCLASS          |
+030 LENGTH(0048) ==> ALL BYTES CONTAIN X'40'.
+060    40404040                                      |                |

 KEY = F000     LENGTH = 000256 (0100)
+000    C1C4C4E4    E2C5D940    C7E4D4C2    E8404040   |ADDUSER GUMBY   |
+010    4040D5C1    D4C54040    40404040    40404D7D   |  NAME        ('|
+020    D1968895    40D84B40    C7A49482    A87D5D40   |JOHN Q. GUMBY') |
+030    404040C3    D3C1E4E3    C8404040    4040404D   |   CLAUTH      (|
+040    82818483    9381A2A2    5D40D7C1    E2E2E6D6   |BADCLASS) PASSWO|
+050    D9C44040    40404DA7    A7A75D40    C1C4C4C3   |RD    (XXX) ADDC|
+060    C1E3C5C7    D6D9E840    4040404D    8381A3F1   |ATEGORY    (CAT1|
+070    5D404040    40404040    40404040    40404040   |)               |
+080 LENGTH(0128) ==> ALL BYTES CONTAIN X'40'.
```

Figure 66. An example of a symptom record when a command fails

In this example, the RACF subsystem return code is 8. In the free format component information, the first X'64' byte area contains the diagnostic message ("IKJ56702I INVALID CLASS NAME, BADCLASS") produced by executing the command image. The next four X'100' areas (only 1 in this case) contain the command image itself (`ADDUSER GUMBY NAME('JOHN Q. GUMBY') CLAUTH(BADCLASS) PASSWORD(XXX) ADDCATEGORY(CAT1)`). Note that the actual password value is not recorded.

Variable data recorded by RRSF (IRRSSQ00)

Log records for RRSF (IRRSSQ00) contain a sequence of global resource serialization (ENQ/DEQ) operations that occur if there is a DEQ error.

CSECT IRRSSQ00

- The log records created from what occurred during the file request that ended with the DEQ failure contain:
  - Failing module: IRRSSQ00
  - Service RC: IRRSSQ00 RC
  - Service reason code: DEQ RC
  - The offset into IRRSSQ00 to DEQ request subroutine
  - Additional text: `addr` DEQ Failed `oper rname_len rname`
    - `addr` = Address of node-def-block representing the file in error
    - `oper` = R,E,I,X, or M for read, erase, insert, read next that are read by MAIN node
    - `rname_len` = Length of DEQ rname
    - `rname` = Text of DEQ rname
- This is followed by one or more log records that contain all of the ENQ/DEQ activity for the request. Each of these contain:
  - Failing module: IRRSSQ00
  - Service RC: IRRSSQ00RC
  - Service reason code: DEQ_RC
  - The offset into IRRSSQ00 where this LOGREC is being printed
  - Additional text is any one of the following:
    - `oper` ENQ qname= SYSZRAC3, rLen `len`, rname !`rname`!, rc=`rc`

      `oper` = Shared or Exclusive
      `len` = Length of rname
      `rname` = The rname, delimited by ! !
    - `rc` RC from ENQ service

      An attempt was made to get the specified ENQ with RC, indicating the results of the attempt.
    - Getting ENQ Not

      No attempt was made to get an ENQ.
    - Rname not set properly

      No attempt is made to get an ENQ.
    - `addr` DEQ rname=`rname`

      `addr` = Node definition address
      `rname` = rname DEQ

      About to DEQ the specified rname.
    - `addr` DEQ RC=`rc`

      `addr` = Node definition address
      `rc` = DEQ return code

      DEQ attempted with return code.
    - `addr` DEQ ERROR `rc`, restarting node

      `addr` = Node definition address
      `rc`: DEQ failing return code

Issued when DEQ fails.

- ENQ in IRRSSQ00 is not received. No DEQ for `addr`.

  `addr` = Node definition address

  Internal error. DEQ requested, but IRRSSQ00 is not able to find corresponding ENQ.

- Track `num` ENQ for `addr`

  `addr` = Node definition address
  `num` = Internal tracking number

  ENQ is received and being tracked.

- Unable to Track ENQ

  Code is not able to track the ENQ received.

- Clear ENQ for `addr`

  `addr` = Node definition block

  DEQ is complete and tracking is clear.

- ENQ `num` not found in list

  `num` = Tracking number

  DEQ is complete, but IRRSSQ00 is not able to find tracking for it.

- `addr` rname is `rname`

  `addr` = Node definition block
  `rname` = rname for ENQ/DEQ

  The next ENQ or DEQ uses this rname.

Control block overviews for RACF

This presents the links between control blocks that can be useful when diagnosing problems related to RACF.

RCVT and SAF router vector table

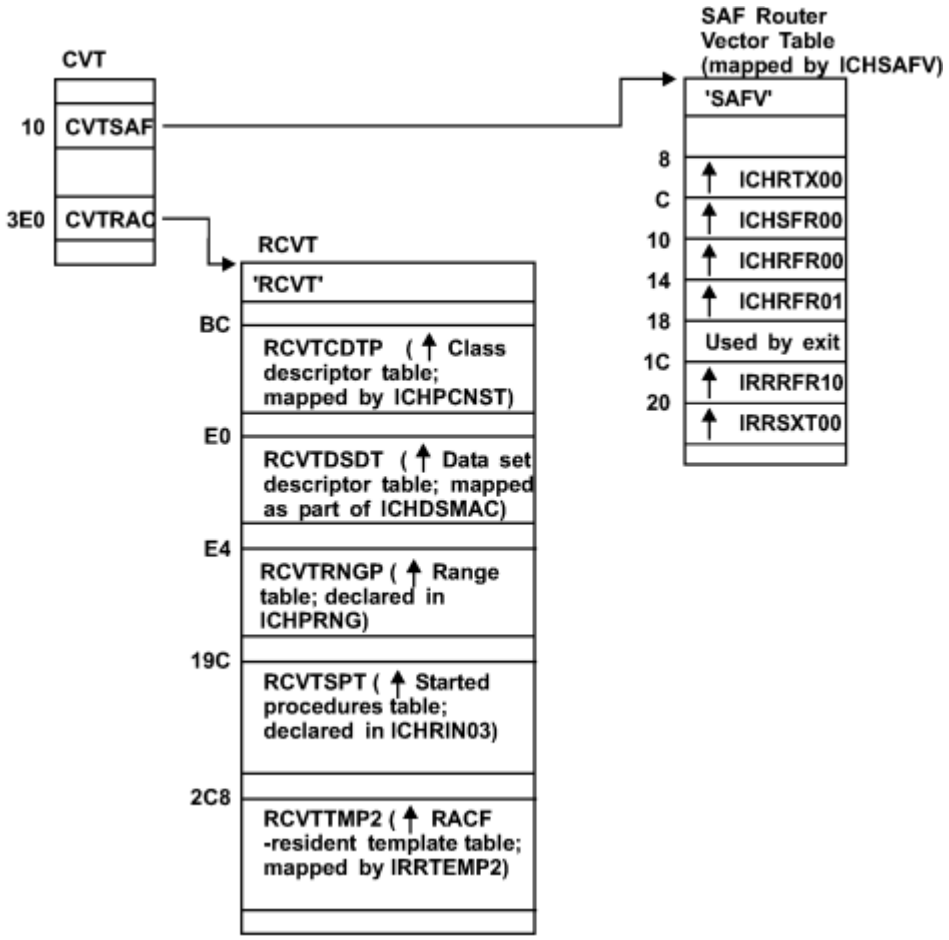


Figure 67. Control block overview: RCVT and SAF router vector table

ACEE and related control blocks

The first time a RACROUTE REQUEST=VERIFY is issued for an address space, there is only one ACEE (accessor environment element), and it is pointed to by the ASXB (see Figure 68 on page 228).

For any subsequent task in the address space (multiple users), the following cases could occur:

- A table of pointers to ACEEs is set up in the address space. This occurs for CICS (Customer Information Control System) or IMS (Information Management System) address spaces.
- Another ACEE is created when the VERIFY request is issued. This ACEE is pointed to only by the TCB for that task. See Figure 69 on page 228.
- When a third-party AUTH request is used (USERID was specified on the RACROUTE REQUEST=AUTH), the ACEE is pointed to by an existing ACEE. See Figure 70 on page 228.

For more information about how ACEEs are created and which ACEE is used, see *z/OS Security Server RACF System Programmer's Guide*.

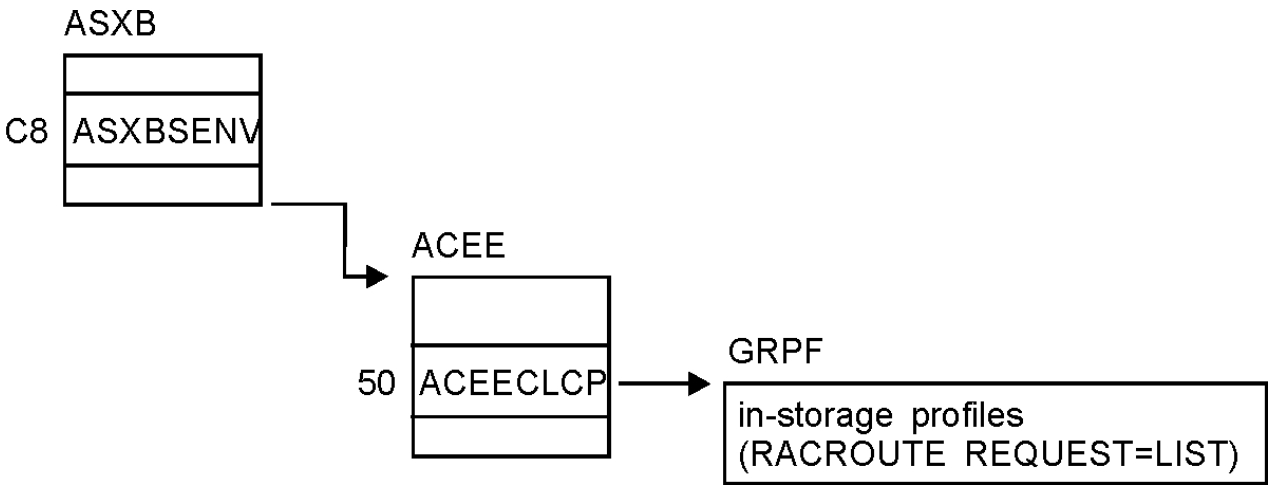


Figure 68. Control block overview: ACEE in a single-user address space

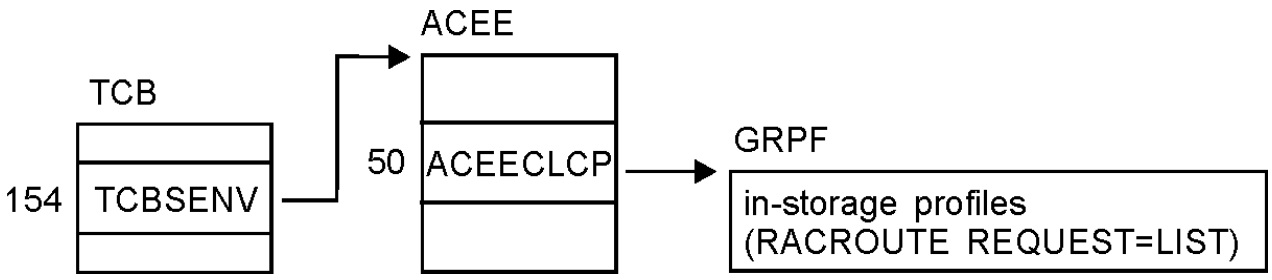


Figure 69. Control block overview: ACEE in a multiple-user address space

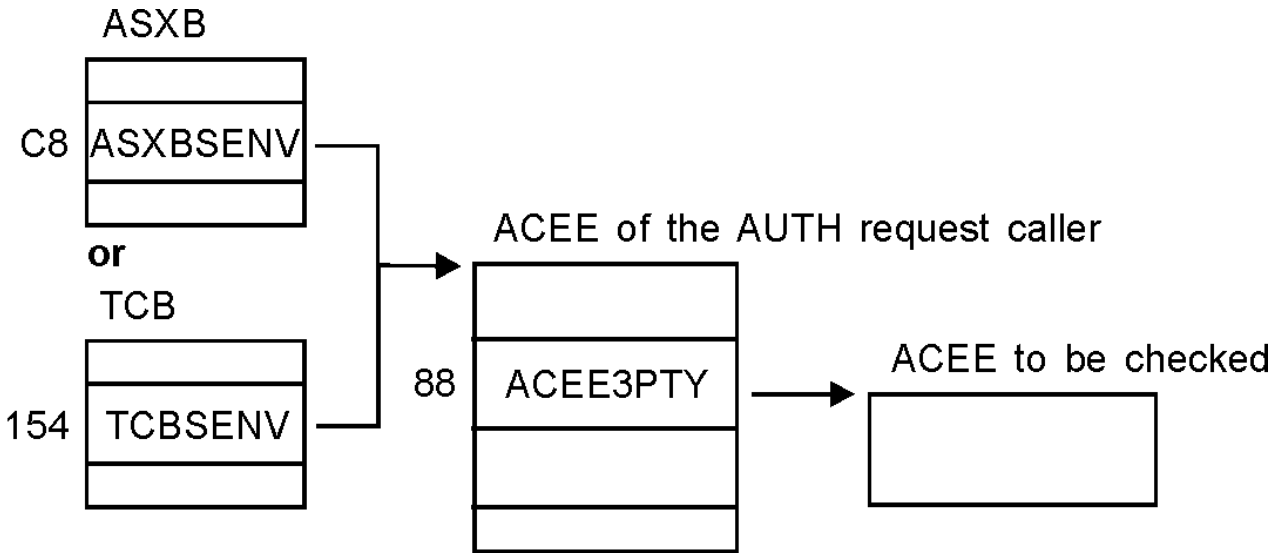


Figure 70. Control block overview: ACEEs when third-party AUTH request is used

SVC dump title descriptions

This describes the SVC (supervisor call instruction) dumps generated by RACF recovery routines.

SVC failure (ICHRST00)

This SVC dump issued by RACF has the form:

ICHRST00-RACF SVCS, ABEND CODE=*sss-rrr*, SVC=*svcname*, USER=RACF *user ID* or *job name*, GROUP=RACF *group ID* or *step name*[, EXIT= *csectname*]

Component:
Resource Access Control Facility (RACF), 5752-XXH00

Issuing Module:
ICHRST00-ESTAE

Explanation:
An abend occurred during processing of one of the RACF SVCs or during processing of the GENLIST or RACLIST operand of the SETROPTS command. The values in the system dump title are:

sss
System completion code (abend code), in hexadecimal.

rrr
Reason code associated with the system completion code (described in *z/OS Security Server RACF Data Areas* in the z/OS Internet library (www.ibm.com/servers/resourcelink/svc00100.nsf/pages/zosInternetLibrary)).

svcname
Name of the RACF routine handling the SVC and issuing the abend.

RACF *user ID* or *job name*
If interactive, RACF user ID of user who encountered the problem.

If batch, then name of job that encountered the problem.

RACF *group ID* or *step name*
If interactive, RACF group of user who encountered the problem.

If batch, then name of step that encountered the problem.

csectname
Name of the CSECT that probably caused the error.

System Action:
The executing task is terminated. The areas dumped are PSA, RGN, LPA, TRT, CSA, and SQA.

Problem Determination:

1. Look for the EXIT shown in the dump title:
   - If the EXIT shown in the system dump title is an installation exit, the installation exit probably caused the error (for a description of installation exits, see *z/OS Security Server RACF System Programmer's Guide*). Debug the installation exit. You can use standard debugging methods to analyze the problem described in this system dump.
   - If the EXIT shown in the system dump title is an IBM routine, or if EXIT is missing from the system dump title, follow the remaining steps in this procedure.
2. Look for message ICH409I as issued for this system dump. See the message text for the following:
   - Abend code and reason code (must be the same as in the dump title)
   - RACF macro or SETROPTS command option (GENLIST or RACLIST) that was being processed
   - Whether parameter validation or other processing was being done

   See *z/OS Security Server RACF Messages and Codes* for an explanation of ICH409I and of the abend code (and its reason code, if supplied).
3. If ICH409I is unavailable, see *z/OS Security Server RACF Messages and Codes* for an explanation of the abend code (*sss*) and reason code (*rrr*) from the dump title.
4. If the explanation of the code does not help you solve the problem and the EXIT named in the system dump title is an IBM routine, see "Abend troubleshooting table" on page 9 for instructions on debugging the problem.

Other Problem Data Produced:
A software record is written to SYS1.LOGREC and includes:

SDWAMODN
Main CSECT (control section) name of SVC (supervisor call instruction)

SDWACSCT
Blanks or name of routine in control at time of error

SDWAREXN
ICHRST00 (recovery routine)

SDWAGR15
Reason code if the abend is a RACF abend

SDWACRC
Completion code

SDWACID
XXH00

SDWAEAS
1 if SDUMP is generated by ICHRST00

SDWAREQ
Zero if SDUMP is generated by ICHRST00

RACF initialization failure (ICHSEC02)

The form of this dump issued by RACF depends on the cause of the dump.

Component:
Resource Access Control Facility (RACF), 5752-XXH00

Issuing Module:
ICHSEC02-ESTAE

Explanation:
An abend occurred during RACF initialization processing. The areas dumped are SQA, CSA, NUC, and RGN.

Problem Determination:

1. See message ICH505A for the abend code that caused the system dump.
2. If message ICH529I was issued, an RVARY command failed.
3. See *z/OS Security Server RACF Messages and Codes* for an explanation of the abend code and the messages.
4. If the explanations in *z/OS Security Server RACF Messages and Codes* do not help you solve the problem, see "Abend troubleshooting table" on page 9 for instructions on debugging the problem.

Other Problem Data Produced:
A software record is written to SYS1.LOGREC and includes:

SDWAMODN
ICHSEC00 (module detecting the error)

SDWAREXN
ICHSEC02 (recovery routine)

SDWACID
XXH00

SDWACSCT
ICHSEC00

SDWAEAS
1 if SDUMP is generated by ICHSEC00

SDWAREQ
Zero if SDUMP is generated by ICHSEC00

RACLIST failure (IRRCOP10)

This dump issued by RACF has the form:

IRRCOP10-RACF RACLIST TO A DATA SPACE FAILED FOR CLASS=*classname*

Component:
Resource Access Control Facility (RACF), 5752-XXH00

Issuing Module:
IRRCOP10-Copy RACLISTed Class to a Data Space

Explanation:
One of the following failed when attempting to copy the in-storage profiles into a data space during RACF initialization:

- SETROPTS RACLIST
- SETROPTS RACLIST REFRESH
- RACROUTE REQUEST=LIST,GLOBAL=YES request
- RACLIST

System Action:
The RACLIST for *classname* failed. A dump was taken to help determine the cause of the problem. Areas dumped are CSA, LSQA, and the data space.

Problem Determination:

1. Determine the cause of the failure by checking the return and reason codes provided by:
   - Message ICH14058I for the SETROPTS RACLIST or SETROPTS RACLIST REFRESH cases
   - Message ICH546I for the RACLIST during initialization case
   - Message IRRL0002I for the RACROUTE REQUEST=LIST,GLOBAL=YES case
2. Use standard debugging methods to analyze the dump itself. To look at a data space with IPCS, you must know the name of the data space. The name of a data space obtained for RACLIST purposes is “IRR*xxxxx*”, where *xxxxx* is a hex number. The name of the data space in this dump can be obtained by an IPCS command, “listdump select”. Simply scan the command's output for “DSPNAME” and use the resulting name and ASID to access the data space.

RACF COPY RACGLIST failure (IRRRLS12)

The form of this dump is:

IRRRLS12-RACF COPY RACGLIST *classname_nnnnn* PROFILES FOR CLASS=*classname*

Component:
Resource Access Control Facility (RACF), 5752-XXH00

Issuing Module:
IRRRLS12RACF RACGLIST to Data Space Routine

Explanation:
Problems were encountered while trying to read A RACGLIST *classname_nnnnn* profile in order to use the RACLIST results stored in those profiles to build the RACLIST data space for *classname*.

System Action:
In addition to taking this dump, RACF issues IRRL0000I, which identifies the specific profile and the manager return and reason codes for the failure. RACF proceeds to RACLIST *classname* by reading *classname*'s discrete and generic profiles from the database to build the RACLIST results, which are then copied to the RACLIST data space. Areas dumped are SQA, LSQA, PSA, RGN, LPA, TRT, CSA, and SUMDUMP.

Problem Determination:

1. Determine the cause of the failure by checking the return and reason codes provided by message IRRL0000I.

2. Use standard debugging methods to analyze the dump itself.

RACF CREATE RACGLIST failure (IRRRLS13)

The form of this dump is:

IRRRLS13RACF CREATE RACGLIST *classname_nnnnn* PROFILES FOR CLASS=*classname*

Component:
Resource Access Control Facility (RACF), 5752-XXH00

Issuing Module:
IRRRLS13RACF Data Space to RACGLIST Routine

Explanation:
Problems were encountered while trying to write a RACGLIST *classname_nnnnn* profile to the database after building a raclist data space while RACLISTing *classname*.

System Action:
In addition to taking this dump, RACF issues IRRL0000I, which identifies the specific profile and the manager return and reason codes for the failure. RACF attempts to delete all the RACGLIST *classname_nnnnn* profiles. The RACLIST operation successfully built the RACLIST data space for *classname*. Areas dumped are SQA, LSQA, PSA, RGN, LPA, TRT, CSA, and SUMDUMP.

Problem Determination:

1. Determine the cause of the failure by checking the return and reason codes provided by message IRRL0000I.
2. Use standard debugging methods to analyze the dump itself.

RACF DELETE RACGLIST failure (IRRRLS14)

The form of this dump is:

IRRRLS14RACF DELETE RACGLIST *classname_nnnnn* PROFILES FOR CLASS=*classname*

Component:
Resource Access Control Facility (RACF), 5752-XXH00

Issuing Module:
IRRRLS14RACF Delete RACGLIST *classname_nnnnn*

Explanation:
Problems were encountered while trying to delete a RACGLIST *classname_nnnnn* profile from the database.

System Action:
In addition to taking this dump, RACF issues IRRL0000I, which identifies the specific profile and the manager return and reason codes for the failure. If the command being processed was RDELETE RACGLIST *classname*, RACF terminates command processing. If the command was one of the commands or requests causing RACF to RACLIST *classname*, the RACLIST succeeded, but RACGLIST *classname_nnnnn* was not updated. Areas dumped are SQA, LSQA, PSA, RGN, LPA, TRT, CSA, and SUMDUMP.

Problem Determination:

1. Determine the cause of the failure by checking the return and codes provided by message IRRL0000I.
2. Use standard debugging methods to analyze the dump itself.

RACF data space recovery failure (IRRRLS15)

The form of this dump is:

IRRRLS15RACF RACLIST TO A DATA SPACE RECOVERY FOR CLASS=*classname*

Component:
Resource Access Control Facility (RACF), 5752-XXH00

Issuing Module:
IRRRLS15RACF RACLIST to Data Space Recovery

Explanation:
One of the following abended while processing a data space:

- SETROPTS RACLIST
- SETROPTS RACLIST REFRESH
- RACLIST during initialization
- RACROUTE REQUEST=LIST, GLOBAL=YES

The ICHRST00 dump described in "SVC failure (ICHRST00)" on page 228 is also taken. This dump provides the data space, which is not provided by the ICHRST00 dump.

System Action:
The command or request failed. A dump was taken to determine the cause of the problem. Areas dumped are CSA, LSQA, and the data space.

Problem Determination:

1. Analyze the ICHRST00 dump as described.
2. Use standard debugging methods to analyze this dump. To look at a data space with IPCS, you must know the name of the data space. The name of a data space obtained for RACLIST purposes is "IRR*xxxxx*" where *xxxxx* is a hex number. The name of the data space in the dump can be obtained by an IPCS command, "listdump select". Simply scan the command's output for "DSPNAME" and use the resulting name and ASID to access the data space.

RACF subsystem failures (IRRSSC01, IRRSSS01, and IRRLOG00)

All dumps issued by the RACF subsystem have the form:

DUMP_TITLE=*uuuu subsystem,vvvvvvvv task,www yyyyy module,zzzzzzz segment.*

The appropriate subsystem name, task name, abend code, module name, and segment name are all provided. The numbers following the module name give the compile date for that module.

An example of a RACF subsystem dump title is:

```
DUMP_TITLE=RACF Subsystem,IRRSSC00 task,0C1 abend,
IRRSSM04 92100 module,WAITECB segment.
```

Component:
RACF (5752-XXH00)

Issuing Modules:
IRRSSC01 IRRSSS01

Explanation:
The RACF subsystem issued dumps.

Problem Determination:

1. See *z/OS MVS System Codes* for an explanation of the abend code and the messages.
2. If the explanations in *z/OS MVS System Codes* do not help you solve the problem, see "Abend troubleshooting table" on page 9 for instructions on debugging the problem.
3. When the dump title identifies IRRLOG00 as the failing CSECT, there might be additional diagnostic information available in the SYSOUT and CEEDUMP DDs for the subsystem started task.

Persistent verification failures

Dumps issued by RACF for persistent verification support in RACF's subsystem have the form:

DUMP_TITLE=COMPON=RACF,COMPID=5752XXH00,ISSUER=IRRSGN06,
MODULE=*mmmmmmmm*,ABEND=S0*xxx*,REASON=*yyyyyyyy*

Component:
RACF (5752-XXH00)

Issuing Module:
mmmmmmmm

Explanation:
The RACROUTE REQ=SIGNON service issued a dump.

Problem Determination:

1. Look up the abend and reason code in *z/OS Security Server RACF Messages and Codes*.
2. Use standard debugging methods to analyze the dump.

RACF I/O failures

Dumps issued by the RACF manager when an I/O occurs while processing a RACHECK, RACINIT, RACDEF, or RACF command request have the form:

DUMP TITLE= ISSUER=IRRMER01-RACF SVCS, I/O ERROR,SVC=*xxxxxxxx*,
USER=*yyyyyyyy* ,GROUP=*zzzzzzzz*

where:

xxxxxxxx
is the SVC name

yyyyyyy
is the RACF user's name from the ACEE or job name.

zzzzzzzz
is the RACF user's group name from the ACEE or step name.

An example of a RACF I/O dump title is:

```
DUMP TITLE= ISSUER=IRRMER01-RACF SVCS, I/O ERROR,SVC=ICHRSV00,
            USER=IBMUSER ,GROUP=SYS1
```

Component:
RACF (5752-XXH00)

Issuing Modules:
IRRMER01

Explanation:
An I/O error occurred and RACF issued a Sdump.

RACF index block error

In some cases where message IRR407I is issued, a DAE suppressible dump might be provided as a problem determination aid. This can occur during SVC 132 processing.

DUMP TITLE=COMPON=RACF,COMPID=5752XXH00,ISSUER=ICHRSV00,
MODULE=IRRMLO00,MANAGER RC=4C,RBA=*xxxxxxxxxxxx*

where:

xxxxxxxxxxxx
is the RBA of the index block (which is not valid) that produced message IRR407I.

An example of a RACF Index Block error dump title is:

```
DUMP TITLE=COMPON=RACF,COMPID=5752XXH00,ISSUER=ICHRSV00,
MODULE=IRRMLO00,MANAGER RC=4C,RBA=000000001000
```

Component:
RACF (5752-XXH00)

Issuing Modules:
ICHRSV00

Explanation:
ICHRSV00 takes a DAE suppressible dump in cases where the RACF manager detects an index block that is not valid. In some cases this error occurs because of storage overlay. This dump captures information that might be helpful in debugging the overlay. IRRMLO00 is the RACF manager routine that determines whether conditions exist that call for this dump to be taken.

Dump title description for Callable Services

The following dump title is displayed on the MVS console when a RACF module used for processing a SAF callable service abends:

DUMP_TITLE=ISSUER=*csect*,COMPID=5752-XXH00,COMPON=RACF,CSECT=*aaaaaaaa*,
LVL=*bbbbbbbb*,ABEND=*ssss*-*rrr*,USR=*cccccccc*

Where:

csect
is the name of the csect that issued the dump, either IRRRCOMR or IRRRCHD7

aaaaaaaa
is the name of the RACF module that abended

bbbbbbbb
is the product fmid (that is, HRF7706)

sss
is the system completion code (abend code), in hexadecimal

rrr
is the reason code associated with the system completion code

cccccccc
is the user's name from the RACF ACEE

When the module abends, the executing task is terminated. The areas dumped are RGN, SQA, TRT, and SUM.

VRA keys provided by RACF for some abends

For some abends, RACF provides VRA (variable recording area) keys that can help in describing a problem to the IBM support center or in narrowing a search when an initial search gets too many matches.

The VRA is part of the SDWA (system diagnostic work area) that recovery routines can use when a problem occurs. The VRA can be included in an SYS1.LOGREC record and in a system dump. To format the VRA, use the DAEDATA control statement in PRDMP or the DAEDATA verbexit in IPCS. Symptoms included in the VRA can be formatted using the SYMPTOMS control statement in PRDMP or the SYMPTOMS verbexit in IPCS.

Figure 71. Sample symptoms for RACF (DAEDATA output)

This figure includes the RETAIN search argument and the VRA keys as formatted by DAEDATA.

```
ICHRST00-RACF SVCS, ABEND CODE=0C1-A62,SVC=ICHRSV00,USER=IBMUSER MODULE SVCDUMP DATE 07/13/87 TIME14.12.06 PAGE 00000004

RETAIN SEARCH ARGUMENT:

RIDS/ICHRSV00#L RIDS/ICHMDR00 AB/S00C1 VALU/H91204074 REGS/0F8B4

REGS/0E78A PRCS/00 000001

SYMPTOMS PRESENT FOR USE AS A UNIQUE DUMP IDENTIFIER BY DAE:

MVS KEY RETAIN SYMPTOM DATA                EXPLANATION
        KEY
------- ------ ------------------------    -----------
MOD/    RIDS/  ICHRSV00                    LOAD MODULE NAME            ┐
CSECT/  RIDS/  ICHMDR00                    ASSEMBLY MODULE CSECT NAME  │
AB/S    AB/S   S00C1                       ABEND CODE-SYSTEM           │
FI/     VALU/H 90EC50580000001091204074    FAILING INSTRUCTION AREA    ├─ Use these symptomsfor initial searches
REGS/   REGS/  0F8B4                       REG/PSW DIFFERENCE          │
REGS/   REGS/  0E78A                       REG/PSW DIFFERENCE          ┘
HRC1/   PRCS/  00000001                    REASON CODE
ADDITIONAL SYMPTOM DATA NOT USED BY DAE TO IDENTIFY THIS DUMP:

MVS KEY RETAIN SYMPTOM DATA       EXPLANATION
        KEY
------- ------ ------------       -----------
VHID/   PCSS/  RACFCNTS ─┐  [1]   HEADER FOR DATA ┐
VHID/   PCSS/  IPLPTR ──┐│        HEADER FOR DATA ├─ Use these symptoms
VHID/   PCSS/  ACTNPTR  ││        HEADER FOR DATA ┘  to analyze a problem
VHID/   PCSS/  TESTPTR  ││        HEADER FOR DATA
VHID/   PCSS/  MGRREQST ││        HEADER FOR DATA ├─ Use the value of MGRREQST to
VHEX/   VALU/H 01000000 ◄┘│       HEX ERROR DATA      narrow a search if your
VHEX/   VALU/H 0000EB74◄──┘       HEX ERROR DATA      first search gets too many
ASID1/  VALU/H 000F               TASK RELATED ASID   matches
ORRC1/  PRCS/  00000001           ORIGINAL REASON CODE
```

1
A number of headers for VRA keys are produced by RACF. These headers (labeled HEADER FOR DATA) describe error data (labeled HEX ERROR DATA). For example, in this dump RACFCNTS is the header for 01000000, and IPLPTR is the header for 0000EB74.

Headers
: **Description**

RACFCNTS
: Number of the current RACF macro request on which the RACF manager failed. This number can be interpreted as X'wwxxyyzz', where:

ww
: Is the number of the IPL being processed.

xx
: Is the number of the action being processed.

yy
: Is the number of the test being processed.

zz
: Is reserved.

IPLPTR
: Pointer to the input parameter list for the current RACF manager request.

ACTNPTR
: Pointer to the action being performed at the time of the abend.

TESTPTR
: Pointer to the test being performed at the time of the abend.

MGRREQST
: RACF manager request ID for the ICHEINTY macro being performed at the time of the abend. This request ID is also reported in message ICH413I. This ID is included only on internal RACF requests.

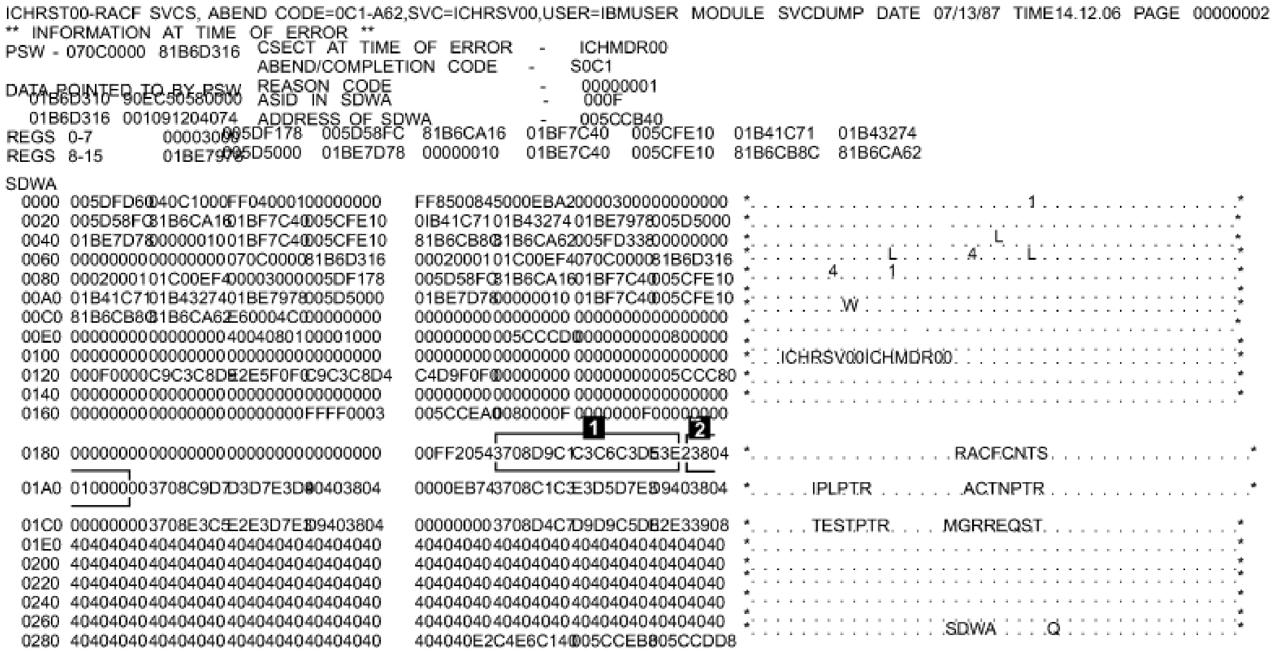
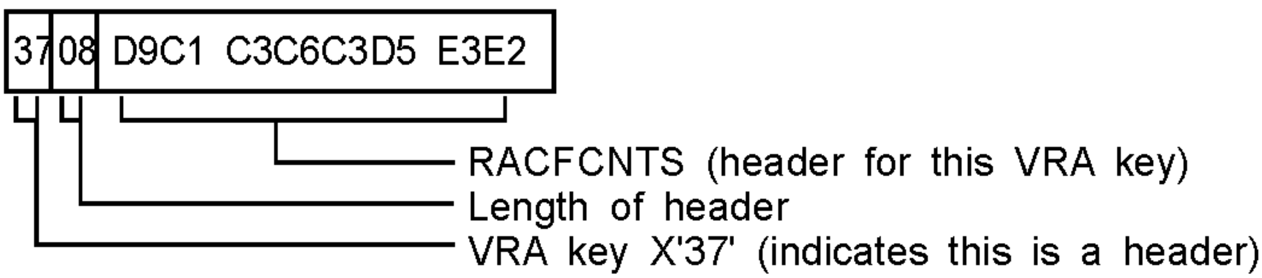
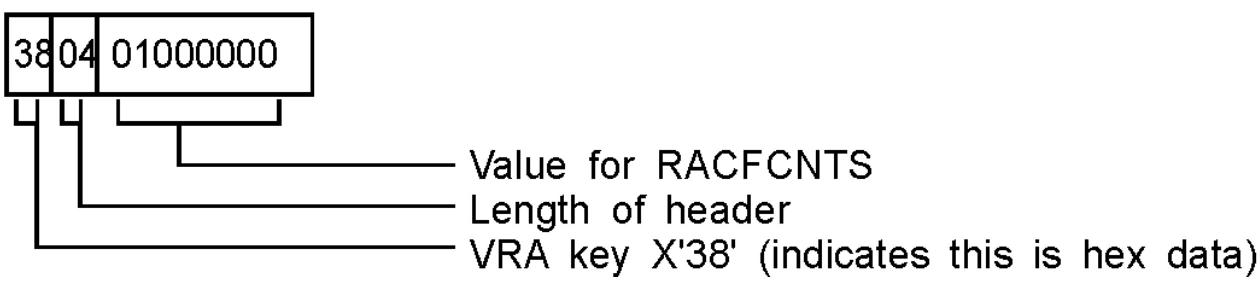

ICHRST00-RACF SVCS, ABEND CODE=0C1-A62,SVC=ICHRSV00,USER=IBMUSER MODULE SVCDUMP DATE 07/13/87 TIME14.12.06 PAGE 00000002
** INFORMATION AT TIME OF ERROR **
PSW - 070C0000 81B6D316
CSECT AT TIME OF ERROR - ICHMDR00
ABEND/COMPLETION CODE - S0C1
DATA POINTED TO BY PSW
REASON CODE - 00000001
01B6D310 90EC5058000
ASID IN SDWA - 000F
01B6D316 001091204074
ADDRESS OF SDWA - 005CCB40
REGS 0-7 00003000 005DF178 005D58FC 81B6CA16 01BF7C40 005CFE10 01B41C71 01B43274
REGS 8-15 01BE7978 005D5000 01BE7D78 00000010 01BE7C40 005CFE10 81B6CB8C 81B6CA62
SDWA
0000 005DFD60 040C1000 FF040001 00000000 FF850084 5000EBA2 00003000 00000000
0020 005D58FC 81B6CA16 01BF7C40 005CFE10 01B41C71 01B43274 01BE7978 005D5000
0040 01BE7D78 00000010 01BF7C40 005CFE10 81B6CB8C 81B6CA62 005FD338 00000000
0060 00000000 00000000 070C0000 81B6D316 00020001 01C00EF4 070C0000 81B6D316
0080 00020001 01C00EF4 00003000 005DF178 005D58FC 81B6CA16 01BF7C40 005CFE10
00A0 01B41C71 01B43274 01BE7978 005D5000 01BE7D78 00000010 01BF7C40 005CFE10
00C0 81B6CB8C 81B6CA62 E60004C0 00000000 00000000 00000000 00000000 00000000
00E0 00000000 00000000 40040801 00001000 00000000 005CCCD0 00000000 00800000
0100 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0120 000F0000 C9C3C8D9 E2E5F0F0 C9C3C8D4 C4D9F0F0 00000000 00000000 005CCC80
0140 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0160 00000000 00000000 00000000 FFFF0003 005CCEA0 0080000F 0000000F 00000000
1
2
0180 00000000 00000000 00000000 00000000 00FF2054 3708D9C1 C3C6C3D5 E3E23804
01A0 01000000 3708C9D7 D3D7E3D9 40403804 0000EB74 3708C1C3 E3D5D7E3 D9403804
01C0 00000000 3708E3C5 E2E3D7E3 D9403804 00000000 3708D4C7 D9D9C5D8 E2E33908
01E0 40404040 40404040 40404040 40404040 40404040 40404040 40404040 40404040
0200 40404040 40404040 40404040 40404040 40404040 40404040 40404040 40404040
0220 40404040 40404040 40404040 40404040 40404040 40404040 40404040 40404040
0240 40404040 40404040 40404040 40404040 40404040 40404040 40404040 40404040
0260 40404040 40404040 40404040 40404040 40404040 40404040 40404040 40404040
0280 40404040 40404040 40404040 40404040 404040E2 C4E6C140 005CCEB8 005CCDD8
ICHRSV00ICHMDR00
RACFCNTS
IPLPTR
ACTNPTR
TESTPTR
MGRREQST
SDWA

Figure 72. Sample SDWA for RACF

1 HEADER FOR DATA



2 HEX ERROR DATA



Front-ending RACF commands

The RACF common command exit, IRREVX01, allows customers to customize or cancel RACF commands before the RACF command processor is invoked. This exit is the recommended vehicle for preprocessing RACF commands. Before RACF provided this exit, some customers created their own front-end processors to preprocess RACF commands. Because RACF 2.2 changed the way RACF processes commands, customers who wrote front-end processors before RACF 2.2 and who want to continue to use these front-end processors might have to change them. This information is intended for those customers.

Note: If you currently use a front-end processor for commands, it is recommended that you replace it with an IRREVX01 exit.

Command processing changes

Before RACF 2.2, when a RACF TSO command was issued, control was passed directly to the appropriate command processor.

To make this happen, RACF included an entry similar to the following for ADDGROUP in the linkage editor statements for each RACF command.

```
INCLUDE AOSBN(ICHCAG00)
INCLUDE AOSBN(other modules)
  :
ALIAS AG,ADDGROUP
ENTRY ICHCAG00
SETCODE AC(1)
NAME ICHCAG00(R)
```

Note: AG or ADDGROUP is the alias name by which the command is issued by a user and ICHCAG00 is the full name of the command by which it is invoked internally.

In RACF 2.2 and OS/390 Security Server (RACF), the entry in the linkage editor statements for most RACF commands look like this:

```
INCLUDE AOSBN(ICHCAG00)
INCLUDE AOSBN(other modules)
  :
ENTRY ICHCAG00
NAME ICHCAG00(R)
```

Note: The aliases are not present.

The aliases are removed from each individual command and placed in a new entry for IRRENV00.

```
INCLUDE AOSBN(IRRENV00)
INCLUDE AOSBN(other modules)
ENTRY IRRENV00
ALIAS AD,ADDSD
ALIAS AG,ADDGROUP
ALIAS other commands
  :
SETCODE AC(1)
NAME IRRENV00(R)
etc.
```

IRRENV00 receives control before each command processor. If the AT or ONLYAT keywords were not specified, IRRENV00 invokes the appropriate command processor using its full name. When the command processor returns control, IRRENV00 determines if automatic direction is required for the command. If so, a work item is sent to the RACF subsystem address space. If the AT or ONLYAT keywords were specified, IRRENV00 sends the command to the RACF subsystem address space for processing.

Considerations if writing a command front-end processor

If IRRENV00 processing is bypassed by a front-end processor, RACF remote sharing functions (RRSF) does not work. When writing a command front-end processor, ensure that IRRENV00 gets control somewhere in the processing before the command processor.

To ensure that all RRSF functions are processed as specified, commands must be invoked by one of the alias names with which they are shipped (for example, AG, ADDGROUP, ADDSD, and so on). When IRRENV00 receives control, the command buffer must have one of the two alias names in the first position. For example, AG or ADDGROUP should be coded, not ICHCAG00. If a front-end processor gets control before IRRENV00, set the command buffer correctly before attaching or linking to IRRENV00. If it gets control after IRRENV00, do not change the content of the command buffer as doing so might result in the information not being processed correctly by automatic command direction.

If the alias is changed to point to a front-end processor, it gets control before IRRENV00 when a command is issued as a TSO command. However, the preprocessor does not get control in all situations. For example, it does not get control for commands resulting from automatic command direction, or for commands issued from the RACF parameter library.

One possible approach to avoid the pitfalls mentioned is to include your front-end processor in RACF's load module.

```
INCLUDE mylib(mymod)
INCLUDE SYSLMOD(ICHCAG00)
ENTRY mymod
NAME ICHCAG00(R)
```

After completing its processing, the front-end processor will need to call the entry point for RACF's command processor. If the front-end processor needs to communicate with the user, be aware that the preprocessor could be running in the RACF subsystem address space and not in the TSO user's address space. The ACEERASP bit (x'40') is set in the ACEEFLG3 flag field of the ACEE for the RACF subsystem address space.

Assigning the entry point to the front-end processor (mymod) causes it to be invoked at the same point in the processing that the specific command processing would be invoked by RACF. Because the alias is not included, the RACF remote sharing functions operate and IRRENV00 is invoked.

The DEBUG command

The DEBUG command starts the diagnostic print statements that wrap the cryptographic library calls and displays additional diagnostic information for RACDCERT invoked ICHEINTY ALTER, RACROUTE REQUEST=EXTRACT, and RACROUTE REQUEST=DEFINE failures. However, the contents of these messages are not documented in any publication. When a problem is encountered, customers can use this keyword to gather diagnostic information for the IBM Support Center.

Chapter 7. Error recovery for RACF sysplex data sharing

This topic describes actions to take in order to allow RACF to continue servicing requests when you encounter a situation where coupling facility recovery scenarios do not work properly.

Actions to recover from a coupling facility error

Do not issue SETXCF to force the rebuild of a structure into a coupling facility that is not available to the system because the result is read-only mode. If SETXCF was issued, you need to exit out of read-only mode by issuing RVARY DATASHARE. Therefore, RACF returns to the original coupling facility.

If you encounter a situation where coupling facility recovery scenarios do not work properly, the following information might make it possible for RACF to continue servicing requests.

For example, assume a sysplex with 3 members: J90, J91, and J92. A coupling facility containing RACF Structure IRRXCF00_B001 has been lost. Member J90 remains active.

The following messages are received at the operator console:

```
IRRX016I RACF MEMBER J90 DETECTED A COUPLING FACILITY ERROR
IXC521I REBUILD FOR STRUCTURE IRRXCF00_B001 HAS BEEN STARTED
IRRX020I REBUILD FOR STRUCTURE IRRXCF00_B001 ON MEMBER J90 HAS BEEN INITIATED
```

The following message is not received at the operator console:

```
IRRX008I REBUILD FOR STRUCTURE IRRXCF00_B001 HAS BEEN COMPLETED
```

Issuing the following command from the operator console:

```
DISPLAY XCF,STRUCTURE
```

displays the following: `IRRXCF00_B001 ALLOCATED REBUILDING`

Issuing the following command from the operator console:

```
DISPLAY GRS,CONTENTION
```

displays the following: `SYSZRAC2 , minor name backup-racf-db is held on system J90 by RACFDS`

In this situation, members of the sysplex might be unable to function properly because RACF is holding enqueues. A rebuild of a RACF structure has been requested but cannot go ahead. The following steps might be helpful in allowing RACF to continue operating, though it will be at a degraded mode.

1. Make sure if message IXC402D has been received one or more times at the operator's console, you reply "down" to all of them. If this allows REBUILD to complete, you do not need to continue with the following steps.
2. Issue the following command at the operator console:

   ```
   SETXCF STOP,REBUILD,STRNAME=IRRXCF00_B001
   ```

   This command stops REBUILD and release enqueues. Additionally, message IRRX004A is received, which displays the following: `IRRX004A MEMBER J90 IS IN READ-ONLY MODE.`
3. Issue the following command at the operator console:

   ```
   RVARY NODATASHARE
   ```

All remaining sysplex members now operate off the database, without the coupling facility. Note that performance will not be as good while running without the coupling facility.

4. You might be able to further improve the situation if you have configured your sysplex appropriately. This means that you have more than one coupling facility and that after the failure of one of them is still available. To illustrate this, read the following:

   Take, for example, two coupling facilities. All structures for the primary RACF database are assigned to one coupling facility and all structures for the backup RACF database are assigned to the other coupling facility. (In this example, assume that no alternate coupling facilities have been assigned.) If you lose one or the other of the coupling facilities in this configuration, you can still get back into data sharing mode, though it will be without a backup database.

   For example:

   - If the coupling facility containing the primary database structures goes down, issue the command:

     ```
     RVARY SWITCH
     ```

     which makes the backup database primary and deactivates the old primary database.
   - If the coupling facility containing the backup database structures goes down, issue the command:

     ```
     RVARY INACTIVE
     ```

     on the backup database.
   - Issue the command:

     ```
     RVARY DATASHARE
     ```

     which allows remaining sysplex members the ability to connect to all structures on the available coupling facility.

   The fourth step has an adverse consequence, however. Although these steps improve performance while a coupling facility is unavailable, your backup and primary databases will most likely become out of sync. This must be resolved before you can go back to normal operations with both primary and backup databases. This can be done by using IRRUT200, as documented in *z/OS Security Server RACF System Programmer's Guide*.

Appendix A. The Inventory Control Block (ICB)

The ICB is included here to help you diagnose RACF problems.

Common Name:	RACF Database Inventory Control Block
Macro ID:	ICHPICB
DSECT Name:	INVICB
Owning Component:	Resource Access Control Facility (XXH00)
Eye-Catcher ID:	RACFICB Offset: 1010 Length: 8
Storage Attributes:	**Subpool:** 231 **Key:** 0 **Residency:** ECSA
Size:	4096 bytes
Created by:	IRRMIN00 or IRRUT400 on the RACF database. In storage by RACF initialization when the system is in data sharing mode or if the database is not shared.
Pointed to by:	DSDEHDR in the DSDT data area
Serialization:	For the copy on the data set: SYSZRACF RESERVE or SYSZRACF ENQ For the in-storage copy: SYSZRACF ENQ
Function:	The ICB is the first record of the RACF database. It provides a control area for the routines that manage RACF data. This control area contains information needed to locate data related to the templates.

Offset (Dec)	Offset (Hex)	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE	4096	INVICB	Inventory Control Block
0	(0)	CHARACTER	2599	ICBHDR	
0	(0)	ADDRESS	4	ICBCHAIN	Reserved
4	(4)	SIGNED	4	ICBBAMNO	Number of BAM blocks in data set
8	(8)	BITSTRING	6	ICCIBRBA	RBA of highest level CIB
14	(E)	BITSTRING	6	ICISSRBA	RBA of first block of index sequence set
20	(14)	BITSTRING	6	ICBAMRBA	RBA of first BAM
26	(1A)	BITSTRING	1	ICBFLAGS	Status

Offset (Dec)	Offset (Hex)	Type	Len	Name (Dim)	Description
		1...		ICBEXTND	RACF will no longer process updates to this version. This bit is set by specifying the LOCINPUT parameter of IRRUT400 or IRRDBU00. When ON, no updates will be made to the RACF data set.
		.1..		*	Reserved, do not use
		..1.		ICBEDAT	RESIDENT data blocks option used on some system
		...1		ICBDSFMT	1 = new RACF data set format
		.... 1...		ICBWIP	HPCS 1 = Write In Progress
		.... .111		*	Reserved
27	(1B)	ADDRESS	1	ICTMPCNT	Number of templates
28	(1C)	BITSTRING	6	ICBAMHWM	BAM HWM.
34	(22)	CHARACTER	16	ICBTEMP (9)	Space for 9 template DEFNS
34	(22)	SIGNED	2	ICTMPL	Template length
36	(24)	ADDRESS	1	ICTMPN	Template number
37	(25)	BITSTRING	1	ICTMRSV1	Align to half word boundary
38	(26)	BITSTRING	6	ICTMPRBA	RBA of FDT
44	(2C)	CHARACTER	6	ICTMRSV2	Reserved
178	(B2)	CHARACTER	16	*	Reserved
194	(C2)	CHARACTER	1	ICBSTAT	Status
		1...		*	Reserved, do not use. Maps the 'RCVTRNA' flag
		.1..		ICBNLS	Bypass RACINIT statistics
		..1.		ICBNDSS	Bypass data set statistics
		...1		ICBNTVS	No tape volume statistics
		.... 1...		ICBNDVS	No direct access volume statistics
		.... .1..		ICBNTMS	No terminal statistics
		.... ..1.		ICBNADS	No ADSP protection

Offset (Dec)	Offset (Hex)	Type	Len	Name (Dim)	Description
		1		ICBEGN	EGN support in effect
195	(C3)	BITSTRING	1	ICBEXTA	Reserved
196	(C4)	CHARACTER	1	ICBSTAT1	
		1...		ICBTAPE	Tape volume protection
		.1..		ICBDASD	DASD volume protection
		..1.		ICBDGEN	Generic profiles for data set class
		...1		ICBDGCM	Generic commands for data set class
		.... 1...		ICBRDSN	Input data set name will be used for logging and messages
		.... .1..		ICBJXAL	JES-XBMALLRACF is in effect
		.... ..1.		ICBJCHK	JES-EARLYVERIFY is in effect
		1		ICBJALL	JES-BATCHALLRACF is in effect
197	(C5)	CHARACTER	1	ICBAUOP	RACF audit options
		1...		*	Reserved
		.1..		ICBAGRO	Audit GROUP class
		..1.		ICBAUSE	Audit USER class
		...1		ICBADAT	Audit DATASET class
		.... 1...		ICBADAS	Audit DASDVOL class
		.... .1..		ICBATAP	Audit TAPEVOL class
		.... ..1.		ICBATER	Audit TERMINAL class
		1		ICBAOPR	Audit OPERATIONS attribute
198	(C6)	BITSTRING	1	ICBAXTA	Reserved
199	(C7)	CHARACTER	1	ICBEROP	Miscellaneous options
		1...		ICBTERP	Terminal authorization checking

Offset (Dec)	Offset (Hex)	Type	Len	Name (Dim)	Description
		.1..		ICBTUAC	Default UACC for terminals not defined to RACF. If ON, UACC = NONE. If OFF, UACC = READ.
		..1.		ICBAVIO	Do not create log record for command violations only
		...1		ICBSAUD	Do not audit SPECIAL users
		.... 1...		ICBMGDG	MODEL-GDG in effect
		.... .1..		ICBMUSR	MODEL-USER in effect
		.... ..1.		ICBMGRP	MODEL-GROUP in effect
		1		ICBLGRP	LIST-OF-GRPS in effect
200	(C8)	SIGNED	4	ICBCHCT (10)	Change count array for all levels of index blocks
240	(F0)	ADDRESS	1	ICBPINV	Global max password interval value, valid range 1-254
241	(F1)	BITSTRING	4	ICBCSTA	Class STATISTICS mask
245	(F5)	BITSTRING	4	ICBCAUD	Class AUDIT mask
249	(F9)	BITSTRING	4	ICBCPRO	Class PROTECTION mask
253	(FD)	UNSIGNED	1	ICBPHIST	Password HISTORY value
254	(FE)	UNSIGNED	1	ICBPRVOK	Password REVOKE value
255	(FF)	UNSIGNED	1	ICBPWARN	Password WARNING value
256	(100)	CHARACTER	10	ICBPSYN (8)	Password SYNTAX rules
256	(100)	UNSIGNED	1	ICBPSLEN	Starting length value
257	(101)	UNSIGNED	1	ICBPELEN	Ending length value
258	(102)	CHARACTER	8	ICBPRULS	Content rules
258	(102)	CHARACTER	1	ICBPRUL1	Content rule
259	(103)	CHARACTER	1	ICBPRUL2	Content rule
260	(104)	CHARACTER	1	ICBPRUL3	Content rule
261	(105)	CHARACTER	1	ICBPRUL4	Content rule
262	(106)	CHARACTER	1	ICBPRUL5	Content rule
263	(107)	CHARACTER	1	ICBPRUL6	Content rule
264	(108)	CHARACTER	1	ICBPRUL7	Content rule
265	(109)	CHARACTER	1	ICBPRUL8	Content rule
336	(150)	UNSIGNED	1	ICBINACT	INACTIVATE interval

Offset (Dec)	Offset (Hex)	Type	Len	Name (Dim)	Description
337	(151)	BITSTRING	4	ICBCGEN	Generic profile checking, class mask
341	(155)	BITSTRING	4	ICBCGCM	Generic commands processing, class mask
345	(159)	BITSTRING	4	ICBMOPT	Miscellaneous options
		1...		ICBFPDS	Fastpath for DATASET class
		.1..		ICBTDSN	Tape data set protection in effect
		..11 1111		*	Reserved
		1...		ICBPRO	PROTECT-ALL is in effect
		.1..		ICBPROF	1 - PROTECT-ALL WARNING is in effect, 0 - PROTECT-ALL FAILURE is in effect (This flag is ignored if ICBPRO has a value of '0'B)
		..1.		ICBEOS	ERASE-ON-SCRATCH is in effect
		...1		ICBEOSL	ERASE-ON-SCRATCH by SECLEVEL is in effect (This flag is ignored if ICBEOS has a value of '0'B)
		.... 1...		ICBEOSA	ERASE-ON-SCRATCH for all data sets is in effect (This flag is ignored if ICBEOS has a value of '0' B)
		.... .1..		ICBPLC	Mixed case password
		…. ..1.		ICBPSC	Special characters are allowed in passwords
		…. ...1			Reserved
		1...		ICBPROG	PROGRAM CONTROL is in effect
		.111 1111			Reserved
		1111 1111		*	Reserved
347	(15B)	BITSTRING	1	*	Reserved
349	(15D)	BITSTRING	4	ICBFPTH	Class FASTPATH mask
353	(161)	UNSIGNED	1	ICBSLVL	Security Level for ERASE-ON-SCRATCH
354	(162)	UNSIGNED	2	ICBRETP	Security Retention Period
356	(164)	UNSIGNED	1	ICBQLLN	Length of prefix for single level data set names

Offset (Dec)	Offset (Hex)	Type	Len	Name (Dim)	Description
357	(165)	CHARACTER	9	ICBQUAL	Installation controlled prefix for single level data set names, plus period for level
366	(16E)	UNSIGNED	1	ICBSLAU	SECLEVEL to audit
367	(16F)	UNSIGNED	1	ICBALIAS	367 (16F) UNSIGNED 1 ICBALIAS application identity mapping byte. 4 stages: • 0=stage 0 – not active – only mapping profiles • 1=stage 1 – active – maintain alias index, but also maintain and use VLF and mapping profiles – Reject locates of application identities • 2=stage 2 – active – maintain and use alias index, but also maintain VLF the mapping profiles – Allow locates of application identities • 3=stage 3 – active – maintain and use only alias index – mapping profiles have been removed
368	(170)	SIGNED	4	ICBCHCTD	Change count for data blocks, used to maintain current in-storage blocks with a shared RACF data set
372	(174)	CHARACTER	8	ICBSWPW	Password for RVARY SWITCH command
380	(17C)	CHARACTER	8	ICBINPW	Password for RVARY ACTIVE/INACTIVE command
388	(184)	BITSTRING	16	ICBLRCL	CDT-ANCHORED RACLISTED profile class mask, ON if active
404	(194)	BITSTRING	16	ICBLGNL	CDT-ANCHORED GENLISTED profile class mask, ON if active
420	(1A4)	BITSTRING	16	ICBLSTAC	Class STATISTICS option mask long version, ON if active
436	(1B4)	BITSTRING	16	ICBLAUDC	Class AUDITING option mask long version, ON if active
452	(1C4)	BITSTRING	16	ICBLPROC	Class PROTECTION option mask long version, ON if active
468	(1D4)	BITSTRING	16	ICBLGENC	Class mask for GENERIC PROFILE CHECKING long version, ON if active

Offset (Dec)	Offset (Hex)	Type	Len	Name (Dim)	Description
484	(1E4)	BITSTRING	16	ICBLGCMC	Class mask for GENERIC COMMAND CHECKING long version, ON if active
500	(1F4)	BITSTRING	16	ICBLFPTC	Class FASTPATH option mask long version, ON if active
516	(204)	SIGNED	1	ICBPMIN	SETROPTS PASSWORD(MINCHANGE) setting
517	(205)	ADDRESS	1	ICNTMPCT	Number of new templates
518	(206)	CHARACTER	16	ICBTMPVR	Template version levels
518	(206)	CHARACTER	8	ICBTMPRL	Release level
526	(20E)	CHARACTER	8	ICBTMPAL	APAR level
534	(216)	BITSTRING	6	ICBTXRBA	RBA of the template extensions
540	(21C)	SIGNED	2	ICBTXLN	Length of the template extension
542	(21E)	CHARACTER	136	*	Reserved
678	(2A6)	BITSTRING	4	ICBVMSSP	VM 370 SYNC counter
682	(2AA)	BITSTRING	4	ICBVMSXA	VMXA ACI SYNC counter
686	(2AE)	CHARACTER	8	ICBBMAPA	VM XA profile name which contains the audit settings
694	(2B6)	CHARACTER	8	ICBBMAPP	VM XA profile name contains the project settings
702	(2BE)	CHARACTER	92	ICBBMAP	Saved copy of the ACI bit map located in the CP nucleus
702	(2BE)	BITSTRING	2	ICBMAPSZ	Length of the entire ACI bit map with uncompressed events
704	(2C0)	BITSTRING	1	ICBDIAL	Indicator for: DIAL CMND
705	(2C1)	BITSTRING	1	ICBPRMSG	Indicator for: DIAL CMND, PRELOGON MESSAGE CMD
706	(2C2)	BITSTRING	88	ICBEVENT	ACI bit map events
794	(31A)	SIGNED	2	ICBSINT	LU Session Interval
796	(31C)	BITSTRING	16	ICBLGAL	SETROPTS "LOGOPTIONS ALWAYS" class mask, ON if active
812	(32C)	BITSTRING	16	ICBLNVR	SETROPTS "LOGOPTIONS NEVER" class mask, ON if active
828	(33C)	BITSTRING	16	ICBLGSU	SETROPTS "LOGOPTIONS SUCCESSES" class mask, ON if active
844	(34C)	BITSTRING	16	ICBLGFL	SETROPTS "LOGOPTIONS FAILURES" class mask, ON if active
860	(35C)	CHARACTER	8	ICBJSYS	User ID from the SETROPTS command JES(NJEUSERID(*id*))
868	(364)	CHARACTER	8	ICBJUND	User ID from the SETROPTS command JES(UNDEFINEDUSER(*id*))

Offset (Dec)	Offset (Hex)	Type	Len	Name (Dim)	Description
876	(36C)	SIGNED	4	ICBRCGSN	RACF group data synchronization value
880	(370)	BITSTRING	1	ICBLOGD	LOGOPTIONS for DATASET
		1...		ICBDLGA	LOGOPTIONS "ALWAYS" for the DATASET class
		.1..		ICBDLGN	LOGOPTIONS "NEVER" FOR THE DATASET CLASS
		..1.		ICBDLGS	LOGOPTIONS "SUCCESSES" for the DATASET class
		...1		ICBDLGF	LOGOPTIONS "FAILURES" for the DATASET class
		 1111		*	Reserved
881	(371)	BITSTRING	1	ICBFLG2X	RACF SETROPTS options
		1...		ICBCMPM	SETROPTS COMPATMODE, ON if active
		.1..		ICBMLSF	SETROPTS MLS FAILURES/WARNING - FAILURES if "ON" - (1) - WARNING if "OFF" - (0)
		..1.		ICBMLAF	SETROPTS MLACTIVE FAILURES/WARNING - FAILURES if "ON" - (1) - WARNING if "OFF" - (0)
		...1		ICBCATF	SETROPTS CATDSNS FAILURES/WARNING - FAILURES if "ON" - (1) - WARNING if "OFF" - (0)
		.... 1...		ICBAAPL	SETROPTS APPLAUDIT ON if active
		.... .1..		ICBNOADC	SETROPTS NOADDCREATOR - on if NOADDCREATOR in effect
		11		*	Reserved
882	(372)	CHARACTER	8	ICBSPAUD	VM/SP profile names which contains the audit settings
890	(37A)	CHARACTER	8	ICBSPCTL	VM/SP profile names which contains the protect settings
898	(382)	BITSTRING	1	ICBFLG2	RACF 1.9.0 SETROPTS options
		1...		ICBSLCL	SETROPTS SECLABELCONTROL - ON if active
		.1..		ICBCATD	SETROPTS CATDSNS - ON if active
		..1.		ICBMLQT	SETROPTS MLQUIET - ON if active

Offset (Dec)	Offset (Hex)	Type	Len	Name (Dim)	Description
		...1		ICBMLST	SETROPTS MLSTABLE - ON if active
		.... 1...		ICBMLS	SETROPTS MLS - ON if active
		.... .1..		ICBMLAC	SETROPTS MLACTIVE - ON if active
		.... ..1.		ICBGNOW	SETROPTS GENERICOWNER - ON if active
		1		ICBAUSL	SETROPTS SECLABELAUDIT - ON if active
899	(383)	CHARACTER	3	ICBNL1	PRIMARY LANGUAGE default
902	(386)	CHARACTER	3	ICBNL2	SECONDARY LANGUAGE default
905	(389)	BITSTRING	1	ICBDIRT	Dirty bits to indicate whether fields have been initialized.
		1...		ICBSESS	SESSIONINTERVAL dirty bit, 0=>Field not initialized, 1=>Field is initialized.
		.111 1111		*	Reserved
906	(38A)	BITSTRING	1	ICBML2F	SETROPTS options for RACF z/OS multilevel security support
		1...		ICBMLFS	Security labels required for File System Objects SETR MLFSOBJ
		.1..		ICBMLIP	Security labels required for IPC Objects SETR MLIPCOBJ
		..1.		ICBMLNM	Display of Names Restricted by Security label SETR MLNAMES
		...1		ICBSBYS	Security label by System requested SETR SECLBYSYSETM
		 1111		*	Reserved
907	(38B)	CHARACTER	85	*	Reserved
992	(3E0)	CHARACTER	6	ICBALRBA	Highest ALIAS index block
998	(3E6)	CHARACTER	6	ICBASRBA	ALIAS index sequence set
1004	(3EC)	ADDRESS	4	ICBSMCT	Count field to control purge of VLF class IRRSMAP
1008	(3F0)	CHARACTER	1	ICBKRBLV	SETROPTS KERBLVL setting

Offset (Dec)	Offset (Hex)	Type	Len	Name (Dim)	Description
1009	(3F1)	CHARACTER	1	*	Reserved
1010	(3F2)	CHARACTER	8	ICBID	Identifier, gets filled in during IRRMIN00 initialization
1018	(3FA)	CHARACTER	6	ICBVRMN	Version/release/modification number, also filled in IRRMIN00
					End of first 1024 bytes
1024	(400)	SIGNED	2	ICTSEGLN	Length of segment table
1026	(402)	BITSTRING	6	ICTSEGRB	RBA of segment table
1032	(408)	BITSTRING	1	ICBRDSFL	RDS flags
		`1... ....`		ICBINITF	ICB was completely initialized by RDS IRRMIN00
		`.111 1111`		*	Reserved
1033	(409)	CHARACTER	3	*	Reserved; Prevent byte skip msg
1036	(40C)	UNSIGNED	4	ICBUSCT	ACEE data repository change count for user profile
1040	(410)	UNSIGNED	4	ICBGPCT	ACEE data repository change count for group profile
1044	(414)	UNSIGNED	4	ICBGRCT	ACEE data repository change count for general resource classes APPL, APPCPORT, JESINPUT, CONSOLE, TERMINAL, GTERMINL & SECLABEL
1048	(418)	ADDRESS	4	ICBUMCT	count field to control purge of VLF class IRRUMAP
1052	(41C)	ADDRESS	4	ICBGMCT	count field to control purge pf VLF class IRRGMAP
1056	(420)	CHARACTER	7	ICBTMPLV	Template level value filled in by IRRMIN00.
1063	(427)	BITSTRING	128	ICBVPROC	Class protection mask
1191	(4A7)	BITSTRING	128	ICBVGENC	Generic profile checking class mask
1319	(527)	BITSTRING	128	ICBVGCMC	Generic command processing class mask
1447	(5A7)	BITSTRING	128	ICBVRCL	CDT-anchored RACLISTed profile class mask
1575	(627)	BITSTRING	128	ICBVGNL	CDT-anchored GENLISTed profile class mask
1703	(6A7)	BITSTRING	128	ICBVFPTC	Fastpath option class mask
1831	(727)	BITSTRING	128	ICBVAUDC	Auditing option class mask
1959	(7A7)	BITSTRING	128	ICBVSTAC	Statistics option mask
2087	(827)	BITSTRING	128	ICBVLGA	SETROPTS LOGOPTIONS ALWAYS class mask
2215	(8A7)	BITSTRING	128	ICBVLNV	SETROPTS LOGOPTIONS NEVER class mask

Offset (Dec)	Offset (Hex)	Type	Len	Name (Dim)	Description
2343	(927)	BITSTRING	128	ICBVLGS	SETROPTS LOGOPTIONS SUCCESSES class mask
2471	(9A7)	BITSTRING	128	ICBVLGF	SETROPTS LOGOPTIONS FAILURES class mask
2599	(A27)	ADDRESS	1	ICTMPXCT	Number of template extensions
2600	(A28)	UNSIGNED	2	ICBPREP	Password algorithm repetition factor
2602	(A2A)	UNSIGNED	2	ICBPMEM	Password algorithm memory factor
2604	(A2C)	UNSIGNED	1	ICBPALG	Password algorithm in effect: 0 = DES or the algorithm as indicated by ICHDEX01 (masking, DES, or installation-defined) 1 = PBKDF2
2605	(A2D)	CHARACTER	1496	ICBRSVD	RESERVED
End of ICB mapping					

Constants

Len	Type	Value	Name	Description
8	CHARACTER	RACFICB	ICBIDC	Constant for ICBID field
6	CHARACTER	1090	ICB1090	Constant representing FMID HRF1902 for ICBVRMN field
6	CHARACTER	1092	ICB1092	Constant representing FMID JRF1922 for ICBVRMN field
6	CHARACTER	2010	ICB2010	Constant representing FMID HRF2210 for ICBVRMN field
6	CHARACTER	2020	ICB2020	Constant representing FMID HRF2220 for ICBVRMN field
6	CHARACTER	2030	ICB2030	Constant representing FMID HRF2230 for ICBVRMN field
6	CHARACTER	2040	ICB2040	Constant representing FMID HRF2240 for ICBVRMN field
6	CHARACTER	2060	ICB2060	Constant representing FMID HRF2260 for ICBVRMN field
6	CHARACTER	2608	ICB2608	Constant representing HRF2608 for ICBVRMN field
6	CHARACTER	7703	ICB7703	Constant representing HRF7703 for ICBVRMN field
6	CHARACTER	7705	ICB7705	Constant representing HRF7705 for ICBVRMN field
6	CHARACTER	7706	ICB7706	Constant representing HRF7706 for ICBVRMN field

Len	Type	Value	Name	Description
6	CHARACTER	7707	ICB7707	Constant representing HRF7707 for ICBVRMN field
6	CHARACTER	7708	ICB7708	Constant representing HRF7708 for ICBVRMN field
6	CHARACTER	7709	ICB7709	Constant representing HRF7709 for ICBVRMN field
6	CHARACTER	7720	ICB7720	Constant representing HRF7720 for ICBVRMN field
6	CHARACTER	7730	ICB7730	Constant representing HRF7730 for ICBVRMN field
6	CHARACTER	7740	ICB7740	Constant representing HRF7740 for ICBVRMN field
6	CHARACTER	7750	ICB7750	Constant representing HRF7750 for ICBVRMN field
6	CHARACTER	7760	ICB7760	Constant representing HRF7760 for ICBVRMN field
6	CHARACTER	7770	ICB7770	Constant representing HRF7770 for ICBVRMN field
6	CHARACTER	7780	ICB7780	Constant representing HRF7780 for ICBVRMN field
6	CHARACTER	7790	ICB7790	Constant representing HRF7790 for ICBVRMN field
6	CHARACTER	HRF77A0	ICB77A0	Constant representing HRF77A0 for ICBVRMN field
6	CHARACTER	HRF77B0	ICB77B0	Constant representing HRF77C0 for ICBVRMN field
6	CHARACTER	HRF77C0	ICB77C0	Constant representing HRF77C0 for ICBVRMN field
6	CHARACTER	HRF77D0	ICB77D0	Constant representing HRF77D0 for ICBVRMN field

Cross reference

Name	Hex Offset	Hex Value	Level
ICBAAPL	371	08	4
ICBADAS	C5	08	4
ICBADAT	C5	10	4
ICBAGRO	C5	40	4
ICBALIAS	16F		3
ICBALRBA	3E0		3
ICBAMHWM	1C		3

Name	Hex Offset	Hex Value	Level
ICBAMRBA	14		3
ICBAOPR	C5	01	4
ICBASRBA	3E6		3
ICBATAP	C5	04	4
ICBATER	C5	02	4
ICBAUOP	C5		3
ICBAUSE	C5	20	4
ICBAUSL	382	01	4
ICBAVIO	C7	20	4
ICBAXTA	C6		3
ICBBAMNO	4		3
ICBBMAP	2BE		3
ICBBMAPA	2AE		3
ICBBMAPP	2B6		3
ICBCATD	382	40	4
ICBCATF	371	10	4
ICBCAUD	F5		3
ICBCGCM	155		3
ICBCGEN	151		3
ICBCHAIN	0		3
ICBCHCT	C8		3
ICBCHCTD	170		3
ICBCMPM	371	80	4
ICBCPRO	F9		3
ICBCSTA	F1		3
ICBDASD	C4	40	4
ICBDGCM	C4	10	4
ICBDGEN	C4	20	4
ICBDIAL	2C0		4
ICBDIRT	89		3
ICBDLGA	370	80	4
ICBDLGF	370	10	4
ICBDLGN	370	40	4
ICBDLGS	370	20	4

Name	Hex Offset	Hex Value	Level
ICBDSFMT	1A	10	4
ICBEDAT	1A	20	4
ICBEGN	C2	01	4
ICBEOS	15A	20	4
ICBEOSA	15A	08	4
ICBEOSL	15A	10	4
ICBEROP	C7		3
ICBEVENT	2C2		4
ICBEXTA	C3		3
ICBEXTND	1A	80	4
ICBFLAGS	1A		3
ICBFLG2	382		3
ICBFLG2X	371		3
ICBFPDS	159	80	4
ICBFPTH	15D		3
ICBGMCT	41C		3
ICBGNOW	382	02	4
ICBGPCT	410		3
ICBGRCT	414		3
ICBHDR	0		2
ICBID	3F2		3
ICBINACT	150		3
ICBINITF	408	80	4
ICBINPW	17C		3
ICBJALL	C4	01	4
ICBJCHK	C4	02	4
ICBJSYS	35C		3
ICBJUND	364		3
ICBJXAL	C4	04	4
ICBKRBLV	3F0		3
ICBLAUDC	1B4		3
ICBLFPTC	1F4		3
ICBLGAL	31C		3
ICBLGCMC	1E4		3

Name	Hex Offset	Hex Value	Level
ICBLGENC	1D4		3
ICBLGFL	34C		3
ICBLGNL	194		3
ICBLGRP	C7	01	4
ICBLGSU	33C		3
ICBLNVR	32C		3
ICBLOGD	370		3
ICBLPROC	1C4		3
ICBLRCL	184		3
ICBLSTAC	1A4		3
ICBMAPSZ	2BE		4
ICBMGDG	C7	08	4
ICBMGRP	C7	02	4
ICBMLAC	382	04	4
ICBMLAF	371	20	4
ICBMLFS	38A	80	4
ICBMLIP	38A	40	4
ICBMLNM	38A	20	4
ICBMLQT	382	20	4
ICBMLS	382	08	4
ICBMLSF	371	40	4
ICBMLST	382	10	4
ICBML2F	38A		3
ICBMOPT	159		3
ICBMUSR	C7	04	4
ICBNADS	C2	02	4
ICBNDSS	C2	20	4
ICBNDVS	C2	08	4
ICBNLS	C2	40	4
ICBNL1	383		3
ICBNL2	386		3
ICBNOADC	371	04	4
ICBNTEMP	206		3
ICBNTMS	C2	04	4

Name	Hex Offset	Hex Value	Level
ICBNTVS	C2	10	4
ICBPELEN	101		4
ICBPHIST	FD		3
ICBPINV	F0		3
ICBPRMSG	2C1		4
ICBPRO	15A	80	4
ICBPROF	15A	40	4
ICBPROG	15B	80	4
ICBPRULS	102		4
ICBPRUL1	102		5
ICBPRUL2	103		5
ICBPRUL3	104		5
ICBPRUL4	105		5
ICBPRUL5	106		5
ICBPRUL6	107		5
ICBPRUL7	108		5
ICBPRUL8	109		5
ICBPRVOK	FE		3
ICBPSLEN	100		4
ICBPSYN	100		3
ICBPWARN	FF		3
ICBQLLN	164		3
ICBQUAL	165		3
ICBRCGSN	36C		3
ICBRDSFL	408		3
ICBRDSN	C4	08	4
ICBRETP	162		3
ICBRSVD	A27		2
ICBSAUD	C7	10	4
ICBSBYS	38A	10	4
ICBSESS	389	80	4
ICBSINT	31A		3
ICBSLAU	16E		3
ICBSLCL	382	80	4

Name	Hex Offset	Hex Value	Level
ICBSLVL	161		3
ICBSMCT	3EC		3
ICBSPAUD	372		3
ICBSPCTL	37A		3
ICBSTAT	C2		3
ICBSTAT1	C4		3
ICBSWPW	174		3
ICBTAPE	C4	80	4
ICBTDSN	159	40	4
ICBTEMP	22		3
ICBTERP	C7	80	4
ICBTMPAL	20E		3
ICBTMPRL	206		3
ICBTMPLV	420		3
ICBTMPVR	206		3
ICBTUAC	C7	40	4
ICBTXLN	21C		3
ICBTXRBA	216		3
ICBUMCT	418		3
ICBUSCT	40C		3
ICBVAUDC	727		3
ICBVFPTC	6A7		3
ICBVGCMC	527		3
ICBVGENC	4A7		3
ICBVGNL	627		3
ICBVLGA	827		3
ICBVLGF	9A7		3
ICBVLGS	927		3
ICBVLNV	8A7		3
ICBVMSSP	2A6		3
ICBVMSXA	2AA		3
ICBVPROC	427		3
ICBVRCL	5A7		3
ICBVRMN	3FA		3

Name	Hex Offset	Hex Value	Level
ICBVSTAC	7A7		3
ICBWIP	1A	08	4
ICCIBRBA	8		3
ICISSRBA	E		3
ICNTMPCT	205		3
ICTMPCNT	1B		3
ICTMPL	22		4
ICTMPN	24		4
ICTMPNN	208		4
ICTMPRBA	26		4
ICBTMPRL	206		4
ICTMRSV1	25		4
ICTMRSV2	2C		4
ICTSEGLN	400		3
ICTSEGRB	402		3
INVICB	0		1

Appendix B. The Security Product Trace Area (SAFT)

The SAFT is included here to help you diagnose RACF problems.

Common Name:	RACF Security Product Trace Area Control Block
Macro ID:	IRRSAFT
DSECT Name:	SAFT
Owning Component:	Resource Access Control Facility (SC1BN)
Eye-Catcher ID:	SAFT Offset: 0 Length: 4
Storage Attributes:	**Subpool:** 245 **Key:** 0 **Residency:** SQA
Size:	200 bytes
Created by:	ICHSEC07 during RACF initialization.
Pointed to by:	SAFVSECT in the SAFV data area
Serialization:	This control block is serialized with SYSZRAC2 ENQ.
Function:	The SAFT provides a communication area for information global to the SAF Trace utility.

Offset (Dec)	Offset (Hex)	Type	Len	Name (Dim)	Description
0	(0)	STRUCTURE		SAFT	IRRSAFT Control Block
0	(0)	CHARACTER	4	SAFTID	SAFT eyecatcher
4	(4)	UNSIGNED	1	SAFTVRSN	Version Number
5	(5)	CHARACTER	3	*	RESERVED
8	(8)	UNSIGNED	1	SAFTSUBP	Storage subpool
9	(9)	UNSIGNED	3	SAFTLENG	Length of SAFT
12	(C)	UNSIGNED	4	SAFTTRCO	Trace routine address
16	(10)	UNSIGNED	4	SAFTTRPC	Trace routine PC number
20	(14)	BITSTRING	4	SAFTTRAC	RACF trace flags
20	(14)	BITSTRING	1	SAFTTACT	RACF trace flags
		`1... ....`		SAFTACTV	A trace is active

Offset (Dec)	Offset (Hex)	Type	Len	Name (Dim)	Description
		.1..		SAFTRRTE	Trace racroute events
		..1.		SAFTMGRT	Trace DB manager events
		...1		SAFTSYSC	Trace system calls
		.... 1...		SAFTSYSZ	Trace SAF3 calls
		.... .1..		SAFTGANC	Trace SET GENERICANCHORs
		11		*	RESERVED
21	(15)	BITSTRING	3	*	RESERVED
24	(18)	BITSTRING	8	SAFTRRTR	Racroute events
24	(18)	BITSTRING	1	*	
		1...		SAFTRALL	Trace all racroutes
		.1..		SAFTAUTH	SAFPREQT = 1 (Auth)
		..1.		SAFTFAUT	SAFPREQT = 2 (Fastauth)
		...1		SAFTLIST	SAFPREQT = 3 (List)
		.... 1...		SAFTDEFN	SAFPREQT = 4 (Define)
		.... .1..		SAFTVFY	SAFPREQT = 5 (Verify)
		.... ..1.		SAFTEXTR	SAFPREQT = 6 (Extract)
		1		SAFTDIR	SAFPREQT = 7 (Dirauth)
25	(19)	BITSTRING	1	*	
		1...		SAFTTKSR	SAFPREQT = 8 (Token Serv)
		.1..		SAFTVFYX	SAFPREQT = 9 (VerifyX)
		..1.		SAFTTEXT	SAFPREQT = A (Token Serv)
		...1		SAFTTBLD	SAFPREQT = B (Token Bld.)

Offset (Dec)	Offset (Hex)	Type	Len	Name (Dim)	Description
		.... 1...		SAFTTXBR	SAFPREQT = C (Extract BR=Y)
		.... .1..		SAFTAUD	SAFPREQT = D (Audit)
		.... ..1.		SAFTTSTS	SAFPREQT = E (Status)
		1		SAFTSIGN	SAFPREQT = F (Signon)
26	(1A	BITSTRING	1	*	
		1...		SAFTMPX	SAFPREQT = 10 (Tmap Xmem)
		.1..		SAFTTXTX	SAFPREQT = 11 (Text Xmem)
		..11 1111		*	RESERVED
27	(1B)	BITSTRING	5	*	RESERVED
32	(20)	BITSTRING	16	SAFTSCLS	Trace RACF syscalls
32	(20)	BITSTRING	1	*	
		1...		SAFTSALL	Trace all syscalls
		.1..	5	SAFTSIU0	Init USP
		..1.		SAFTSDU0	Delete USP
		...1		SAFTSMF0	Make FSP
		.... 1...		*	RESERVED
		.... .1..		SAFTSMM0	Set File Mode Creation Mask
		.... ..1.		SAFTSKA0	Check Access
		1		SAFTSKP0	Check Privilege
32	(22)	BITSTRING	1	*	
		1...		SAFTSUM0	Get Umap
		.1..		SAFTSGM0	Get Gap
		..1.		SAFTSGG0	Get Supplemental Groups

Offset (Dec)	Offset (Hex)	Type	Len	Name (Dim)	Description
		...1		SAFTSSU0	Set Uid
		.... 1...		SAFTSEU0	Set Effective Uid
		.... .1..		SAFTSSG0	Set Gid
		.... ..1.		SAFTSEG0	Set Effective Gid
		1		SAFTSCO0	Change Owner Group
34	(22)	BITSTRING	1	*	
		1...		SAFTSCF0	Change file mode
		.1..		SAFTSCA0	Change file audit options
		..1.		SAFTSEX0	Exec Set Uid, Gid
		...1		SAFTSAU0	Audit
		.... 1...		SAFTSKO0	Check Process owner
		.... .1..		SAFTSQS0	Query Security options
		.... ..1.		SAFTSQF0	Query File options
		1		SAFTSCS0	Clear SETID
35	(23)	BITSTRING	1	*	
		1...		SAFTSKF0	Check File Owner
		.1..		SAFTSMR0	Make Root FSP
		..1.		SAFTSPT0	PTRACE Authority Check
		...1		SAFTSUG0	Get User groups
		.... 1...		SAFTSFK0	Fork Exit
		.... .1..		SAFTSMI0	Make ISP

Offset (Dec)	Offset (Hex)	Type	Len	Name (Dim)	Description
		.... ..1.		SAFTSKI0	Check IPC
		1		SAFTSCI0	IPC Control
36	(24)	BITSTRING	1	*	
		1...		SAFTSC20	Check Owner 2 Files
		.1..		SAFTSGE0	Get EUid/EGid SupGrp
		..1.		SAFTSDI0	R_dceinfo
		...1		SAFTSDK0	R_dcekey
		.... 1...		SAFTSUD0	R_dceruid
		.... .1..		SAFTSDA0	R_dceauth
		.... ..1.		SAFTSIA0	InitAcee
		1		SAFTSEQ0	RACF Admin Request
37	(25)	BITSTRING	1	*	
		1...		SAFTSIM0	User Map
		.1..		SAFTSDL0	R_datalib
		..1.		SAFTSMK0	Kerb
		...1		SAFTSPK0	Ticket Serve
		.... 1...		SAFTSPX0	R_PKIServ
		.... .1..		SAFTSCH0	R_cacheserv
		.... ..1.		SAFTSPY0	R_proxyserv
		1		SAFTSCL0	R_SetFacl
38	(26)	BITSTRING	10	*	RESERVED
48	(30)	BITSTRING	8	SAFTZCLS	Trace SAF3 calls
48	(30)	BITSTRING	1	*	

Offset (Dec)	Offset (Hex)	Type	Len	Name (Dim)	Description
		1...		SAFTZALL	All SAF3 clls
		.1..		SAFTZA00	aznAccess
		..1.		SAFTSC00	aznCreds
		 1111		*	RESERVED
49	(31)	BITSTRING	7	*	RESERVED
56	(38)	BITSTRING	8	SAFTMTRC	Trace DB manager events
56	(38)	BITSTRING	1	*	
		1...		SAFTMLL	All DB manager event reads
		.1..		SAFTMRD	reads
		..1.		SAFTMALT	writes
		...1		SAFTMALI	alteri
		 1111		*	RESERVED
57	(39)	BITSTRING	7	*	RESERVED
64	(40)	ADDRESS	4	SAFTRRPR	Address of pre-function call indicator
68	(44)	ADDRESS	4	SAFTRRPS	Address of post-function call indicator
72	(48)	ADDRESS	4	SAFTSCPR	Address of pre-function call indicator
76	(4C)	ADDRESS	4	SAFTSCPS	Address of post-function call indicator
80	(50)	ADDRESS	4	SAFTZCPR	Address of pre-function call indicator
84	(54)	ADDRESS	4	SAFTZCPS	Address of post-function call indicator
88	(58)	ADDRESS	4	SAFTMGPR	Address of pre-function call indicator
92	(5C)	ADDRESS	4	SAFTMGPS	Address of post-function call indicator
96	(60)	ADDRESS	4	SAFTGAPR	Address of pre-function call indicator

Offset (Dec)	Offset (Hex)	Type	Len	Name (Dim)	Description
100	(64)	ADDRESS	4	SAFTGAPS	Address of post-function call indicator
104	(68)	ADDRESS	4	*	RESERVED
108	(6C)	UNSIGNED	2	*	RESERVED
110	(6E)	UNSIGNED	1	SAFTALLC	List count for allow filter
111	(6F)	UNSIGNED	1	SAFTTOTC	List count total for all filters
112	(70)	ADDRESS	4	SAFTALST	Trace ASID LIST
116	(74)	ADDRESS	4	SAFTJLST	Trace JOBS LIST
120	(78)	ADDRESS	4	SAFTCLST	Trace CLASS LIST
124	(7C)	ADDRESS	4	SAFTULST	Trace USERID LIST
128	(80)	ADDRESS	8	*	RESERVED
136	(88)	CHARACTER	16	SAFTRRFL	Flags to the security product
136	(88)	CHARACTER	8	SAFTRPRE	Pre process flag
144	(90)	CHARACTER	8	SAFTRPOS	Post process flag
152	(98)	CHARACTER	16	SAFTSCFL	Flags to the security product
152	(98)	CHARACTER	8	SAFTSPRE	Pre process flag
160	(A0)	CHARACTER	8	SAFTSPOS	Post process flag
168	(A8)	CHARACTER	16	SAFTZCFL	Flags to the PD product
168	(A8)	CHARACTER	8	SAFTZPRE	Pre process flag
176	(B0)	CHARACTER	8	SAFTZPOS	Post process flag
184	(B8)	CHARACTER	16	SAFTMGFL	Flags to the security product
184	(B8)	CHARACTER	8	SAFTMPRE	Pre process flag
192	(C0)	CHARACTER	8	SAFTMPOS	Post process flag
200	(C8)	CHARACTER	16	SAFTGAFL	Flags to security product
200	(C8)	CHARACTER	8	SAFTGPRE	Pre process flag
208	(D0)	CHARACTER	8	SAFTGPOS	Post process flag
End of SAFT mapping					

Appendix C. IPCS formatting for RACF control blocks

This topic explains how to analyze RACF dump information by providing formatting support for RACF control blocks by using IPCS formatting.

IPCS formatting is intended only for diagnosis use when requested by the IBM support center.

Using the VERBX RACFDATA command

Use the VERBX RACFDATA command, without any options, to provide a list of acceptable report options. The following report options are provided:

RRSF
Shows remote requests that are actively being processed on a RACF address space TCB, including any requests that are queued for execution. The output summarizes the requests being processed and provides output containing commands. For example:

```
IP CBF 00067F28 ASID(x'001E') STR(IRRSSTCB) EXIT
```

For more information, cut and past the IP CBF command line to the command input area in the IPCS screen:

```
IPCS OUTPUT STREAM ---------------------------------- Line 0 Cols 1 78
Command ===> IP CBF 00067F28 ASID(x'001E') STR(IRRSSTCB) EXIT SCROLL ===> CSR
```

Press ENTER and you see the control block that you requested. Press PF3 to return to the previous control block.

CDT
Shows class descriptor table information, including static and dynamic class descriptors. Static is the common storage and most frequently available. Dynamic are in a data space and usually not dumped. Both CDTs are pointing to structures that reside in RACF data spaces. The formatting code attempts to locate the data space in the dump, but if it is not present, a message is issued identifying the data space that is missing from the dump.

If there is a problem with CDT processing, request a dump that includes RACF data spaces. The option to use on the DUMP or SLIP command is:

```
             DSPNAME=(1.IRR*)
```

This dumps all data spaces that are owned by ASID(1) that start with IRR.

DSDT
Shows information about all active primary and backup RACF data sets.

DSG
Shows information about requests from other systems in the sysplex.

Using the SUMMARY FORMAT command

When using the SUMMARY FORMAT command, it drives the invocation of exits for control blocks that it formats. The RACF IPCS support provides exits for ASXB, TCB, STCB, and RBs. These exits look for data that is relevant to RACF for the control block. For example, ASXBSENV and TCBSENV might point to an ACEE. When this field is nonzero, then the exit generates output:

```
        IP CBF 009EC300 ASID(X'0067') STR(ACEE) EXIT
```

For more ACEE information, cut and paste this command to the command line and run it. The EXIT option on the CBF command causes an ACEE exit to get control which formats different extensions on the ACEE. For example, you can see the list of groups that the user ID is connected to.

The STCB exit looks for the presence of a RACD and generates a command:

```
        IP CBF 7FFFAD00 ASID(X'0067') STR(IRRPRACD)
```

RB is the most frequent exit. The RB exit determines whether it is an SVRB and if the SVRB was created for SVC 82, 83, 84 or 85 (all the RACF SVCs). For each SVC, there is a SAF parameter list and a function unique parameter list. The RB exit identifies the function being called, who invoked the SVC and CBF commands to format any parameter lists that are present. For example:

```
            The previous RB has called RACF SVC 82 for function RACHECK
            Return address of SAF caller: 00D2A93A
                 ASID(X'0018') D2A93A. IFG0194A+293A IN PLPA
            IP CBF 008D1D40 ASID(X'0018') STR(ICHSAFP)
            IP CBF 008D1D68 ASID(X'0018') STR(ICHACHKL)
```

For more information about parameter lists, see “Parameter list descriptions” on page 151.

The RRSF formatter displays all of the defined RRSF nodes and some of their characteristics. Included might be workspace file information, communication settings, and state.

Using the control block formatters

You can format many of the control blocks RACF uses. See “Using the SUMMARY FORMAT command” on page 269 for examples. Also, see the RCVT, by using the CBF ICHPRCVT command, for another example, which can be useful to find your active exits. A complete list can be found in the results of IPCSDATA CURRENT ACTIVE.

Appendix D. Accessibility

Accessible publications for this product are offered through IBM Documentation (www.ibm.com/docs/en/zos).

If you experience difficulty with the accessibility of any z/OS information, send a detailed message to the Contact the z/OS team web page (www.ibm.com/systems/campaignmail/z/zos/contact_z) or use the following mailing address.

IBM Corporation
Attention: MHVRCFS Reader Comments
Department H6MA, Building 707
2455 South Road
Poughkeepsie, NY 12601-5400
United States

Notices

This information was developed for products and services that are offered in the USA or elsewhere.

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
United States of America

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

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

This information could include missing, incorrect, or broken hyperlinks. Hyperlinks are maintained in only the HTML plug-in output for IBM Documentation. Use of hyperlinks in other output formats of this information is at your own risk.

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 supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
Site Counsel
2455 South Road

Poughkeepsie, NY 12601-5400
USA

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this 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.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

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.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which 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.

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.

Depending upon the configurations deployed, this Software Offering may use session cookies that collect each user's name, email address, phone number, or other personally identifiable information for purposes of enhanced user usability and single sign-on configuration. These cookies can be disabled, but disabling them will also eliminate the functionality they enable.

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 IBM's Privacy Policy at ibm.com®/privacy and IBM's Online Privacy Statement at ibm.com/privacy/details in the section entitled "Cookies, Web Beacons and Other Technologies," and the "IBM Software Products and Software-as-a-Service Privacy Statement" at ibm.com/software/info/product-privacy.

Policy for unsupported hardware

Various z/OS elements, such as DFSMSdfp, JES2, JES3, and MVS, contain code that supports specific hardware servers or devices. In some cases, this device-related element support remains in the product even after the hardware devices pass their announced End of Service date. z/OS may continue to service element code; however, it will not provide service related to unsupported hardware devices. Software problems related to these devices will not be accepted for service, and current service activity will cease if a problem is determined to be associated with out-of-support devices. In such cases, fixes will not be issued.

Minimum supported hardware

The minimum supported hardware for z/OS releases identified in z/OS announcements can subsequently change when service for particular servers or devices is withdrawn. Likewise, the levels of other software products supported on a particular release of z/OS are subject to the service support lifecycle of those

products. Therefore, z/OS and its product publications (for example, panels, samples, messages, and product documentation) can include references to hardware and software that is no longer supported.

- For information about software support lifecycle, see: IBM Lifecycle Support for z/OS (www.ibm.com/software/support/systemsz/lifecycle)
- For information about currently-supported IBM hardware, contact your IBM representative.

Programming interface information

This publication documents information that is NOT intended to be used as a programming interface of z/OS.

Programming Interface Information
End Programming Interface Information

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at Copyright and Trademark information (www.ibm.com/legal/copytrade.shtml).

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle, its affiliates, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft and Windows are trademarks of Microsoft Corporation in the United States, other countries, or both.

Index

Special Characters

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

R

S

T

U

V

W

X

Z

Product Number: 5650-ZOS

GA32-0886-50