z/VM

RACF Security Server Diagnosis Guide

version 6 release 1

GC24-6214-00

z/VM

RACF Security Server Diagnosis Guide

version 6 release 1

GC24-6214-00

Note:
Before using this information and the product it supports, read the information in “Notices” on page 157.

This edition applies to version 6, release 1, modification 0 of IBM z/VM (product number 5741-A07) and to all subsequent releases of this product until otherwise indicated in new editions.

This edition replaces GC24-6145-01.

Contents

Figures vii

Tables ix

About This Document xi
Intended Audience xi
Planning for Problem Diagnosis xii
Where to Find More Information xiii

How to send your comments to IBM xv
If you have a technical problem xv

Part 1. Diagnosing Possible Problems in RACF 1

Chapter 1. Starting Diagnosis 3
Where to Find RACF Information. 3
Troubleshooting the Problem 4
Diagnosis Hints 4
Troubleshooting Tables for Problems in RACF 9
Prerequisites 9
Introduction to Procedure 9
Procedure for Using the Troubleshooting Tables 9
Troubleshooting Table: Abends 10
Troubleshooting Table: Documentation 21
Troubleshooting Table: Incorrect Output 22
Troubleshooting Table: Logon Problems. 28
Troubleshooting Table: Loops 29
Troubleshooting Table: Messages 30
Troubleshooting Table: Performance Problems 32
Troubleshooting Table: Return Codes from RACF Macros 33
Troubleshooting Table: Waits 34
Performing a Preliminary Search for a Matching Problem 34
Prerequisites. 34
Introduction to Procedure 34
Procedure. 35

Chapter 2. Collecting and Analyzing Problem Data 39
Collecting Problem Data 39
Obtaining a Description of a RACF Installation 39
Obtaining a System Dump on z/VM 39
Formatting a System Dump on z/VM 40
Analyzing Console Logs 40
Prerequisites. 40
Introduction to Procedure 40
Procedure. 41
Validating Input 41
Checking the Parameter List Passed to RACF Macros on z/VM 41
Prerequisites. 41
Introduction to Procedure 41
Procedure: 41
Analyzing a Wait Problem on z/VM 42

Chapter 3. Searching for a Matching Problem 43

Prerequisites: 43
Introduction to Procedure 43
Procedure. 43

Chapter 4. Reporting RACF Problems to IBM. 45
Prerequisites. 45
Introduction to Procedure 45
Procedure. 45

Part 2. Diagnosis Reference for RACF 55

Chapter 5. Troubleshooting Your RACF Database 57
Format of the RACF Database 57
Header Block (ICB) 58
Templates. 58
Segment Table Block 59
BAM Blocks 59
Index Blocks. 60
Profiles. 66
Database Unload Utility for a RACF database (IRRDBU00) 69
Block Update Command for a RACF database (BLKUPD) 70
Considerations prior to and When Using the BLKUPD Utility 70
Hierarchy of the BLKUPD Command. 70
BLKUPD Command 71

Chapter 6. Parameter Lists 87
RACROUTE Parameter List 88
AUDIT service parameter list. 91
AUTH service parameter list 92
DEFINE service parameter list 96
DIRAUTH Service Parameter List 102
EXTRACT service parameter list 103
FASTAUTH service parameter list 105
LIST service parameter list 106
STAT service parameter list. 107
TOKENBLD service parameter list 108
TOKENMAP/TOKENXTR service parameter list 112
VERIFY/VERIFYX service parameter list 113
ICHEINTY parameter list 117

Chapter 7. Diagnosis Reference for RACF on z/VM 121
Process Flows for RACF on z/VM 121
Process Flow for a RACF Command Session 121
Process Flow for the RAC Command Processor 122
Process Flows for CP Commands, DIAGNOSE Codes, and System Functions 123
Process Flow for RACF Macros in the RACF Service Machine 126
Process Flow for RACROUTE Macro under CMS. 127
Process Flow for the RACROUTE Macro on z/VM with CMS 128
Process Flow for the RACROUTE Macro on z/VM with GCS 130
Process Flow for RACF Installation Exits on z/VM 131
Process Flow for RACF Utilities on z/VM 132
Process Flow for the RACF Report Writer on z/VM 133
Control Block Overview for RACF on z/VM 134
Problem Determination for a RACF Service Machine 136

Part 3. Appendixes 143

Appendix. The ICB Data Area 145
ICB. 145
Constants 151
Cross Reference 151

Notices 157

Programming Interface Information 159

Trademarks 161

Glossary 163

Bibliography 165
Where to Get z/VM Information 165
z/VM Base Library 165
Overview 165
Installation, Migration, and Service 165
Planning and Administration. 165
Customization and Tuning 165
Operation and Use 165
Application Programming. 165
Diagnosis 166
z/VM Facilities and Features 166
Data Facility Storage Management Subsystem for VM 166
Directory Maintenance Facility for z/VM 166
Open Systems Adapter/Support Facility 166
Performance Toolkit for VM 167
RACF Security Server for z/VM 167
Remote Spooling Communications Subsystem Networking for z/VM 167
Prerequisite Products 167
Device Support Facilities 167
Environmental Record Editing and Printing Program. 167

Index 169

Figures

1. Format of the RACF Database . . . 58
2. Template blocks . . . 58
3. Block Masks of the Database . . . 60
4. Example of a RACF Index for the Database . . . 61
5. Format of the Index Block . . . 62
6. Example of an Index Block Header . . . 63
7. Normal Index Entry for USR26 with a Base, TSO, and DFP Segment . . . 64
8. Entry Data for the Normal Index Entry . . . 65
9. Pointer Index Entry in Level-1 Block . . . 65
10. Example of a Profile Displayed by the BLKUPD Utility . . . 69
11. The Relationship between the BLKUPD Command and Its Levels of Subcommands . . . 71
12. RACROUTE Parameter List . . . 89
13. AUDIT Service Parameter List . . . 91
14. AUTH Service Parameter List . . . 92
15. DEFINE Service Parameter List . . . 96
16. DIRAUTH service parameter list . . . 102
17. EXTRACT Service Parameter List . . . 103
18. FASTAUTH Service Parameter List . . . 105
19. LIST Service Parameter List . . . 106
20. STAT Service Parameter List . . . 107
21. TOKENBLD Service Parameter List . . . 108
22. TOKENMAP/TOKENXTR Service Parameter List . . . 112
23. VERIFY/VERIFYX Service Parameter List . . . 113
24. ICHEINTY Parameter List . . . 117
25. Process Flow for a RACF Command Session . . . 121
26. Process Flow for the RAC Command Processor . . . 122
27. Process Flow for LOGON, AUTOLOG, and XAUTOLOG Commands . . . 123
28. Process Flow for Certain CP Commands, DIAGNOSE Codes, and System Functions . . . 124
29. Process Flow for DIAGNOSE X'A0' . . . 125
30. Process Flow for RACF Macros on z/OS . . . 126
31. Process Flow for the RACROUTE Macro on z/VM with CMS . . . 128
32. Process Flow for the RACROUTE Macro on z/VM with GCS . . . 130
33. Process Flow for RACF Installation Exits on z/VM . . . 132
34. Process Flow for RACF Utilities . . . 133
35. Process Flow for RACF Report Writer on z/VM . . . 134
36. Control Blocks Useful for Diagnosing RACF on z/VM . . . 134
37. Control Block Overview: RCVT and SAF Router Vector Table . . . 135
38. Control Block Overview: ACEE on z/VM . . . 136
39. Problem Determination Chart for RACF z/VM Initialization . . . 137

Tables

1. Determining if You have a RACF Problem 4
2. Index to Troubleshooting Tables for RACF 9
3. Troubleshooting Table: Abends 11
4. Troubleshooting Table: Documentation 21
5. Troubleshooting Table: Incorrect Output 22
6. Troubleshooting Table: Logon Problems 28
7. Troubleshooting Table: Loops 29
8. Troubleshooting Table: Messages 30
9. Troubleshooting Table: Return Codes 32
10. Troubleshooting Table: Return Codes 33
11. Troubleshooting Table: Waits 34
12. Preliminary Search Symptoms for RACF in z/VM 36
13. Examples of Search Arguments for RACF on z/VM 37
14. Additional Search Symptoms for RACF 43
15. Information to Gather before Calling IBM with Abend Problems 46
16. Information to Gather before Calling IBM with Wait Problems 54
17. Segment Identifiers 64
18. Cross-Reference for RACROUTE REQUEST=type, the Independent RACF System Macros, and the Parameter List Name 87
19. List of the Diagnose X'A0' subcodes 125

About This Document

This document contains information about diagnosing problems on the IBM® RACF® Security Server for z/VM®.

Though this document is specific to z/VM, there are references to z/OS®. These references are only applicable when sharing a RACF database with a z/OS system.

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 Dump Viewing Facility
- Understand the externals for RACF.

Before using this document, collect the following problem data:

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

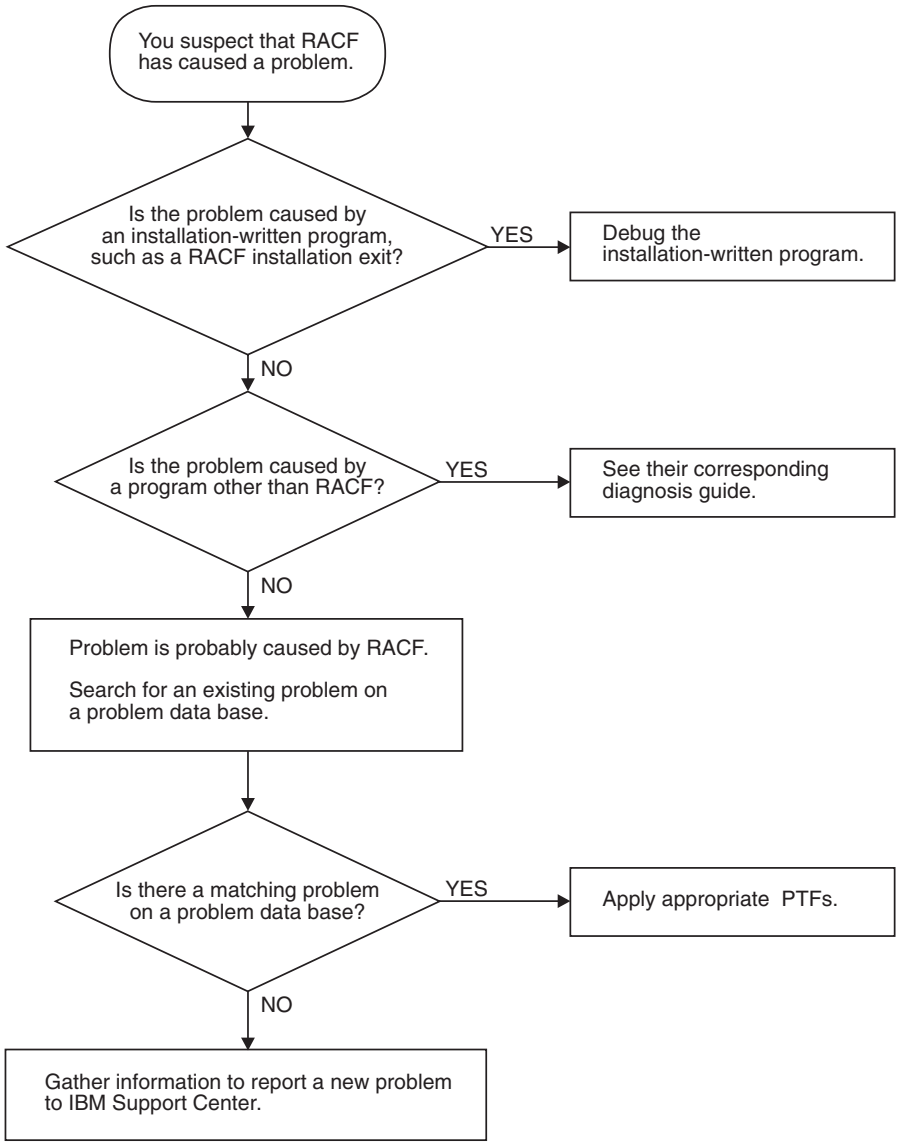
If you do not have this data, see your system diagnosis guide and perform its procedures.

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 product.

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.

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



Planning for Problem Diagnosis

Before using RACF, consider making the following preparations for diagnosis.

- Properly install and operate of RACF so that you get adequate problem data (such as messages and dumps) when problems occur.
- Perform **timely** and **complete** backups of the RACF database.
- Have 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 the system security administrator could also 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.
- Prepare to use the following debugging techniques (at least):

  - Obtain messages that have been sent to the system console or the security console
  - Check the console log of the RACF service machines.
- Prevent common problems by using RACF macros and utilities. See the recommendations in "Common Usage Problems with RACF Macros and Utilities" on page 5.
- Correct any problems that were caused while using RACF profiles and options. See *z/VM: RACF Security Server Security Administrator's Guide* for more information.

Where to Find More Information

For information about related publications, refer to the "Bibliography" on page 165.

Links to Other Online Documents

If you are viewing the Adobe® Portable Document Format (PDF) version of this document, it might contain links to other documents. A link to another document is based on the name of the requested PDF file. The name of the PDF file for an IBM document is unique and identifies the edition. The links provided in this document are for the editions (PDF names) that were current when the PDF file for this document was generated. However, newer editions of some documents (with different PDF names) might exist. A link from this document to another document works only when both documents reside in the same directory.

How to send your comments to IBM

We appreciate your input on this publication. Feel free to comment on the clarity, accuracy, and completeness of the information or give us any other feedback that you might have.

Use one of the following methods to send us your comments:

1. Send an e-mail to mhvrcfs@us.ibm.com
2. Visit the z/VM reader's comments Web page at www.ibm.com/systems/z/os/zvm/zvmforms/webqs.html
3. Mail the comments to the following address:
   IBM Corporation
   Attention: MHVRCFS Reader Comments
   Department H6MA, Mail Station P181
   2455 South Road
   Poughkeepsie, NY 12601-5400
   U.S.A.
4. Fax the comments to us as follows:
   From the United States and Canada: 1+845+432-9405
   From all other countries: Your international access code +1+845+432-9405

Include the following information:

- Your name and address
- Your e-mail address
- Your telephone or fax number
- The publication title and order number:
  z/VM V6R1 RACF Security Server Diagnosis Guide
  GC24-6214-00
- The topic and page number related to your comment
- The text of your comment

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

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

If you have a technical problem

Do not use the feedback methods listed above. Instead, do one of the following:

- Contact your IBM service representative.
- Contact IBM technical support.
- Visit the z/VM support Web page at www.vm.ibm.com/service/
- Visit the IBM mainframes support Web page at www.ibm.com/systems/support/z/

Part 1. Diagnosing Possible Problems in RACF

This part contains information on diagnosing problems in RACF, including step-by-step procedures for collecting and analyzing problem data, searching for a matching problem, and reporting RACF problems to IBM.

Chapter 1. Starting Diagnosis

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

- "Where to Find RACF Information"
- "Troubleshooting the Problem" on page 4
- "Performing a Preliminary Search for a Matching Problem" on page 34.

Where to Find RACF Information

For an overview of RACF concepts, see *z/VM: RACF Security Server General User's Guide*.

For help in gathering information about your RACF installation, see your RACF security administrator and RACF auditor, or see the section on conducting an audit in *z/VM: RACF Security Server Auditor's Guide*.

See Part 2, "Diagnosis Reference for RACF," on page 55 for the following:

- Parameter list descriptions
- Process flows
- Control block overviews
- SVC dump title descriptions
- A description of variable recording area (VRA) keys provided by RACF
- A problem determination flowchart for the RACF service machine on z/VM.

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

For the mappings of RACF control blocks (such as the RCVT and the ACEE), see *z/VM: Security Server RACROUTE Macro Reference*.

Note to the Reader

The section "Troubleshooting and the RACF Database," is in Part 2, "Diagnosis Reference for RACF," on page 55. The skills used in working with the BLKUPD command are different from the skills used in general troubleshooting for suspected RACF problems.

Troubleshooting the Problem

The troubleshooting information for RACF is described in:

- **"Diagnosis Hints."** This section includes:
  - How to make sure 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 9.** 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 34.** This section tells you how to search a database for a matching problem.

Diagnosis Hints

This section 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 helps you make sure 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 the job and operator console log, RACF console log, or other problem related output.

Table 1. Determining if 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.
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. Check that error message handling is ON. This can be done with the command SET EMSG ON (not SET EMSG TEXT). If a batch job encountered the error, check that the batch console was spooled back to the originating user.
Did RACF return control to the program that made the request?	

If RACF was the cause of the problem, the answers to these questions are normally YES or NO. Some exceptions, such as incorrect output, are covered in "Troubleshooting the Problem" on page 4.

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.

Common Usage Problems with RACF Macros and Utilities

This section 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/VM: RACF Security Server System Programmer's Guide*.

IRRUT100 Utility

Error: Assuming 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: Not specifying valid MAP or INDEX options when using z/VM.

Action:

The MAP and/or INDEX options *must* be specified in the RACVERFY FILE file. Also, these options must be specified in separate records in the file. For example:

```
MAP
INDEX
END
```

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

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 invalid 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.

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

Action:

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

Error: Assuming 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 the utility failed. See *z/VM: RACF Security Server 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 are desired 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 may be accessing the RACF database. If any changes are made to the database, the output of the utility will not be reliable.

Action:

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

Error: RACF manager return code X'50' (decimal 80) trying to access the RACF database. This return code means that you attempted to update a RACF database that has had the utility run against it with the LOCKINPUT keyword specified.

Action:

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

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

IRRDBU00 Utility

Error: RACF manager return code X'50' (decimal 80) from other tasks trying to access the RACF database. This return code means that they attempted to update a RACF database while the utility was executing with the LOCKINPUT keyword specified.

Action:

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

Do not try updating until the utility is finished running with LOCKINPUT. Run the utility at a different time when updates are *not* needed, or run it against a copy of the RACF database—not against the active database.

Error: I/O error while reading the database; your data set has slipped into secondary extents. The following 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
```

The following 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.

RACF Macros

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 will not be 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 were ignored.

Action:

If the MACRO macro is continued to 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 will ignore 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 will be 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 desired task (no command or utility that will do 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.

RACHECK Macro

Error: Abend 282-64 (decimal 100) on RACHECK macro with ACEE (accessor environment element) keyword, abend 283-4C (decimal 76) on RACINIT DELETE with ACEE keyword specified.

Action:

Check the specification of the ACEE keyword. On a RACINIT, ACEE specifies a pointer to the address of the ACEE. On a RACHECK, ACEE specifies a pointer to the ACEE.

Error: Return code 4, abend 282-20 (DECIMAL 32), or abend 0C4 during parameter validation on RACHECK 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

Prerequisites

You need to be familiar with RACF's basic concepts and functions in order to use the troubleshooting tables. If you need an overview, see "Where to Find RACF Information" on page 3.

Note: If you need a complete description of RACF concepts and functions, see *z/VM: RACF Security Server General User's Guide* and *z/VM: RACF Security Server System Programmer's Guide*.

Introduction to Procedure

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 information that you will need if you report the problem to IBM.

Procedure for Using the Troubleshooting Tables

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.
2. If you find a matching symptom type:
   a. Go to the page listed in the second column, **page**, in Table 2. 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 11, follow the steps for diagnosing the problem.

Note: 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	Page
Abend	11
Documentation error	21
Incorrect output	
• Improper access to a resource with no message issued	22
• Incorrect messages	23
• Incorrect abends	24
• RACF list commands	25
• Report from DSMON or RACF report writer	25
• SMF records	26
• RACF database	26
• Return or reason code from RACF macro or utility	27
• Return code from RACF report writer	27

Table 2. Index to Troubleshooting Tables for RACF (continued)

Symptom Type	Page
Logon Problems	28
Table 7	29
Troubleshooting Table: Messages	30
Troubleshooting Table: Performance Problems	32
Troubleshooting Table: Return Codes from RACF Macros	33

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

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

   On z/VM, check that error message handling is ON. This can be done with the command SET EMSG ON (not SET EMSG TEXT). If a batch job encountered the error, check that the batch console was spooled back to the originating user.
2. If you find an error message that might relate to the problem, see Table 8 on page 30.
3. If you do not have a system dump for the problem, and you want to request one, see "Obtaining a System Dump on z/VM" on page 39.
4. If you have a system dump for the problem, display or print the symptom record in the dump.
   - Use the SYMP subcommand of the DUMPSCAN command from the dump viewing facility.

   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 9.
5. If you still don't have a symptom type,
   a. Check the general problem analysis advice in your system diagnosis guide.
   b. Read the information in "Making Sure This Is a RACF Problem" on page 4.
   c. If you need additional assistance, report the problem to IBM.

Troubleshooting Table: Abends

Table 3 on page 11 describes abends (abnormal end of task) related to RACF, including:

- **RACF abends,** which are abends that result from using RACF functions that could occur under any operating system.
- **CMS abends,** which are abends issued by CMS or OS simulation of CMS.

The first failure data capture for RACF and CMS OS simulation abends can be improved by enabling a debug option in the RACF virtual machine. Use the following procedure:

1. xedit the RACONFIG EXEC.
   - Change the variable `debug = 'NO'` to `debug = 'YES'`.
   - Save the modified RACONFIG EXEC on RACFVM's 191 disk.
2. Restart the RACFVM service machine.

This setting will cause in RACFVM to dump and restart if an abend SVC is issued which is deemed to be indicative of an error path.

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 means of a RACROUTE REQUEST=AUTH.
- SVC X'83' (in decimal, SVC 131) occurs because a RACINIT macro was invoked independently or by means of a RACROUTE REQUEST=VERIFY.
- SVC X'84' (in decimal, SVC 132) occurs because a RACLIST macro was invoked independently or by means of a RACROUTE REQUEST=LIST.
- SVC X'85' (in decimal, SVC 133) occurs because a RACDEF macro was invoked independently or by means of 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 the following special cases first: **For 0C1 abends:** Verify that the failing load module has been properly link-edited by reviewing the install and service procedures as documented in the *Program Directory for RACF Security Server for z/VM*. **For 0C4 abends:** If the failing module is ICHCLD00, ICHDSM35, IRRRCK00, IRRRIN04, IRRRIN05, IRRRXT00, or or IRRFLC00, it is likely that the module is trying to address a private profile that does not exist. This can happen when a RACHECK preprocessing exit passes a return code of 4 or 8 but does not build a private profile if the private option was specified on the RACHECK. To correct this error, correct the RACHECK preprocessing exit so that it builds a private profile when requested on the RACHECK macro. 5. If these preliminary checks fail, check to see if RACF is the probable cause of the problem or to determine who called the RACF module. **Note:** To check the parameters actually passed to RACF on a macro request, see "Checking the Parameter List Passed to RACF Macros on z/VM" on page 41. Call the IBM support center for assistance in checking the parameters. 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 15 on page 46 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(s) issued with the abend. This could be message IRR401I or ICH409I (the IRR message is followed by a CST message). **Find the abend reason code, if any, in the text for the messages.** 2. See *z/VM: RACF Security Server Messages and Codes* for information about the abend code and the accompanying message (if you haven't looked up the explanation already). In general, the information in *z/VM: RACF Security Server Messages and Codes* should be sufficient to correct the problem. 3. If the information in *z/VM: RACF Security Server 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 34. 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 z/VM" on page 39. 6. Format the system dump. • See "Formatting a System Dump on z/VM" on page 40. 7. You may want to re-create the problem you are investigating (or set up the conditions to allow the problem to reoccur) after setting the trap to obtain the dump. 8. See Chapter 3, "Searching for a Matching Problem," on page 43. 9. If you still do not find a matching problem, see Chapter 4, "Reporting RACF Problems to IBM," on page 45.

Table 3. Troubleshooting Table: Abends (continued)

Symptom	Recommended Procedures for Abends
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(s) issued with the abend. This could be message IRR401I or ICH409I (on z/VM, the IRR message is followed by a CST message). **Find the abend reason code, if any, in the text for the messages.** 2. If you haven't looked up the explanations already, see *z/VM: RACF Security Server 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/VM: RACF Security Server Messages and Codes* should be sufficient to correct the problem. 3. If the information in *z/VM: RACF Security Server 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(s) might indicate conditions that caused the abend. If message IDs were not included with the messages and you can re-create the error, do the following: • If a batch job produced the output, check that the batch console was spooled back to the originating user. If an interactive user had the problem, check that the user had SET EMSG ON (not SET EMSG TEXT). 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. • A CST message, if issued for this abend, probably contains the name of the calling module. 6. 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 book for the program of which the module is a part. To determine the diagnosis or logic book for the program, find the name of the component that owns the module. If the component is owned by IBM, the general information manual for the product containing the component has the name and order number of any diagnosis or logic publication. 7. See "Performing a Preliminary Search for a Matching Problem" on page 34. If you find a matching problem, apply the appropriate PTFs (program temporary fixes). 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 procedure. **Please be sure to fill out a blank copy of the worksheet shown in Table 15 on page 46.** 8. If the abend did not generate a system dump, and the problem is reproducible, obtain a system dump. 9. See Chapter 3, "Searching for a Matching Problem," on page 43. 10. If you still do not find a matching problem, see Chapter 4, "Reporting RACF Problems to IBM," on page 45.

Table 3. Troubleshooting Table: Abends (continued)

Symptom	Recommended Procedures for Abends
Abends: X'382' X'383' X'385'	**Description:** A user exit associated with an SVC (supervisor call instruction) passed an invalid return code. **Action:** 1. See the message(s) issued with the abend. This could be message IRR401I or ICH409I (on z/VM, the IRR message is followed by a CST message). 2. If you haven't looked up the explanations already, see *z/VM: RACF Security Server 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/VM: RACF Security Server Messages and Codes* should be sufficient to correct the problem. 3. If the information in *z/VM: RACF Security Server 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(s) might indicate conditions that caused the abend. If message IDs were not included with the messages and you can re-create the error, do the following: • If a batch job produced the output, check that the batch console was spooled back to the originating user. If an interactive user had the problem, check that the user had SET EMSG ON (not SET EMSG TEXT). 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. A CST message, if issued for this abend, probably contains the name of the calling module. 6. 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 book for the program of which the module is a part. To determine the diagnosis or logic book for the program, find the name of the component that owns the module. If the component is owned by IBM, the general information manual for the product containing the component has the name and order number of any diagnosis or logic publication. 7. See "Performing a Preliminary Search for a Matching Problem" on page 34. If you find a matching problem, apply the appropriate PTFs (program temporary fixes). 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 procedure. **Please be sure to fill out a blank copy of the worksheet shown in Table 15 on page 46.** 8. If the abend did not generate a system dump, and the problem is reproducible, obtain a system dump. See "Obtaining a System Dump on z/VM" on page 39. 9. Format the system dump. See "Formatting a System Dump on z/VM" on page 40. 10. You may want to re-create the problem you are investigating (or set up the conditions to allow the problem to reoccur) after setting the trap to obtain the dump. 11. See Chapter 3, "Searching for a Matching Problem," on page 43. If you still do not find a matching problem, see Chapter 4, "Reporting RACF Problems to IBM," on page 45.

Table 3. Troubleshooting Table: Abends (continued)

Symptom	Recommended Procedures for Abends
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(s) 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 haven't looked up the explanations already, see *z/VM: RACF Security Server 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/VM: RACF Security Server Messages and Codes* should be sufficient to correct the problem. 3. See Chapter 4, "Reporting RACF Problems to IBM," on page 45.
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 34. 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 z/VM" on page 39. 5. Format the system dump. • See "Formatting a System Dump on z/VM" on page 40. 6. You may want to re-create the problem you are investigating (or set up the conditions to allow the problem to reoccur) after setting the trap to obtain the dump. 7. See Chapter 3, "Searching for a Matching Problem," on page 43. 8. If you still do not find a matching problem, see Chapter 4, "Reporting RACF Problems to IBM," on page 45.

Table 3. Troubleshooting Table: Abends (continued)

Symptom	Recommended Procedures for Abends
Abends: X'482' X'483' X'485'	**Description:** RACF manager error during SVC (supervisor call instruction) processing. **Action:** 1. See the message(s) issued with the abend. This could be message IRR401I or ICH409I (on z/VM, the IRR message is followed by a CST message). **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/VM: RACF Security Server Messages and Codes* for information about the abend code and the accompanying message (if you haven't looked up the explanation already). In general, the information in *z/VM: RACF Security Server Messages and Codes* should be sufficient to correct the problem. 4. If the information in *z/VM: RACF Security Server 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 34. 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. 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 z/VM" on page 39. 7. Format the system dump. • See "Formatting a System Dump on z/VM" on page 40. 8. You may want to re-create the problem you are investigating (or set up the conditions to allow the problem to reoccur) after setting the trap to obtain the dump. 9. See Chapter 3, "Searching for a Matching Problem," on page 43. 10. If you still do not find a matching problem, see Chapter 4, "Reporting RACF Problems to IBM," on page 45.
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(s) 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 haven't looked up the explanations already, see *z/VM: RACF Security Server 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/VM: RACF Security Server Messages and Codes* should be sufficient to correct the problem. 3. See Chapter 4, "Reporting RACF Problems to IBM," on page 45.

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, then no ACEE (accessor control environment) 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 haven't looked up the explanations already, see *z/VM: RACF Security Server 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/VM: RACF Security Server Messages and Codes* should be sufficient to correct the problem. 4. If the information in *z/VM: RACF Security Server 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(s) might indicate conditions that caused the abend. If message IDs were not included with the messages and you can re-create the error, do the following: If a batch job produced the output, check that the batch console was spooled back to the originating user. If an interactive user had the problem, check that the user had SET EMSG ON (not SET EMSG TEXT). 6. 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. A CST message, if issued for this abend, probably contains the name of the calling module. 7. 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 book for the program of which the module is a part. To determine the diagnosis or logic book for the program, find the name of the component that owns the module. If the component is owned by IBM, the general information manual for the product containing the component has the name and order number of any diagnosis or logic publication. 8. See "Performing a Preliminary Search for a Matching Problem" on page 34. If you find a matching problem, apply the appropriate PTFs (program temporary fixes). 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 procedure. **Please be sure to fill out a blank copy of the worksheet shown in Table 15 on page 46.** 9. If the abend did not generate a system dump, and the problem is reproducible, obtain a system dump: See "Obtaining a System Dump on z/VM" on page 39. 10. Format the system dump. See "Formatting a System Dump on z/VM" on page 40. 11. You may want to re-create the problem you are investigating (or set up the conditions to allow the problem to reoccur) after setting the trap to obtain the dump. 12. See Chapter 3, "Searching for a Matching Problem," on page 43. If you still do not find a matching problem, see Chapter 4, "Reporting RACF Problems to IBM," on page 45.

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). **Action:** 1. See the message(s) issued with the abend. This could be message IRR401I or ICH409I (followed by a CST message). **Find the abend reason code, if any, in the message text.** 2. See *z/VM: RACF Security Server 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/VM: RACF Security Server Messages and Codes* should be sufficient to correct the problem. 3. If the information in *z/VM: RACF Security Server Messages and Codes* is not sufficient, follow the steps in this procedure. 4. Check if other error messages were issued with the abend messages. These messages might indicate the context in which the abend occurred. Messages preceding the abend message(s) might indicate what caused the abend. If message IDs were not included with the messages, you can re-create the error, and a batch job produced the output, check that the batch console was spooled back to the originating user. If an interactive user had the problem, check that the user had SET EMSG ON (not SET EMSG TEXT). 5. Find the **calling** 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. A CST message, if issued for this abend, probably contains the name of the calling module. 6. 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 book for the module's product. To determine the diagnosis or logic book for the program, find the name of the component that owns the module. If the component is owned by IBM, the general information manual for the product has the name and order number of any diagnosis or logic publication. 7. See "Performing a Preliminary Search for a Matching Problem" on page 34. If you find a matching problem, apply the appropriate PTFs (program temporary fixes). If you do not find a matching problem, you will need to collect problem information to report to IBM, using the steps in this procedure. **Please be sure to fill out a blank copy of the worksheet shown in Table 15 on page 46.** 8. 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 commands or in the ISPF command tables. 9. See "Performing a Preliminary Search for a Matching Problem" on page 34. 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. 10. If the abend did not generate a system dump, and the problem is reproducible, obtain a system dump: See "Obtaining a System Dump on z/VM" on page 39. 11. Format the system dump. See "Formatting a System Dump on z/VM" on page 40. 12. You may want to re-create the problem you are investigating (or set up the conditions to allow the problem to reoccur) after setting the trap to obtain the dump. 13. See Chapter 3, "Searching for a Matching Problem," on page 43. If you still do not find a matching problem, see Chapter 4, "Reporting RACF Problems to IBM," on page 45.

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(s) issued with the abend. **Find the abend reason code, if any, in the text for the messages.** 2. See *z/VM: RACF Security Server Messages and Codes* for information about the abend code and the accompanying message (if you haven't looked up the explanation already). In general, the information in *z/VM: RACF Security Server Messages and Codes* should be sufficient to correct the problem. 3. If the information in *z/VM: RACF Security Server 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 34. 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 z/VM" on page 39. 6. Format the system dump. • See "Formatting a System Dump on z/VM" on page 40. 7. You may want to re-create the problem you are investigating (or set up the conditions to allow the problem to reoccur) after setting the trap to obtain the dump. 8. See Chapter 3, "Searching for a Matching Problem," on page 43. 9. If you still do not find a matching problem, see Chapter 4, "Reporting RACF Problems to IBM," on page 45.

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(s) issued with the abend. This could be message IRR401I or ICH409I (followed by a CST message). **Find the abend reason code, if any, in the text for the messages.** 2. See *z/VM: RACF Security Server 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/VM: RACF Security Server Messages and Codes* should be sufficient to correct the problem. 3. If the information in *z/VM: RACF Security Server Messages and Codes* is not sufficient, follow the steps in this procedure. 4. Check if 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(s) might indicate conditions that caused the abend. If message IDs were not included with the messages and you can re-create the error, do the following: If a batch job produced the output, check that the batch console was spooled back to the originating user. If an interactive user had the problem, check that the user had SET EMSG ON (not SET EMSG TEXT). 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. A CST message, if issued for this abend, probably contains the name of the calling module. 6. 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 book for the program of which the module is a part. To determine the diagnosis or logic book for the program, find the name of the component that owns the module. If the component is owned by IBM, the general information manual for the product containing the component has the name and order number of any diagnosis or logic publication. 7. See "Performing a Preliminary Search for a Matching Problem" on page 34. If you find a matching problem, apply the appropriate PTFs (program temporary fixes). 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 procedure. **Please be sure to fill out a blank copy of the worksheet shown in Table 15 on page 46.** 8. If the abend did not generate a system dump, and the problem is reproducible, obtain a system dump: • See "Obtaining a System Dump on z/VM" on page 39. 9. Format the system dump. • See "Formatting a System Dump on z/VM" on page 40. 10. You may want to re-create the problem you are investigating (or set up the conditions to allow the problem to reoccur) after setting the trap to obtain the dump. 11. See Chapter 3, "Searching for a Matching Problem," on page 43. 12. If you still do not find a matching problem, see Chapter 4, "Reporting RACF Problems to IBM," on page 45.
Abends: X'E82' X'E83' X'E84' X'E85'	**Description:** RACF is not installed. **Action:** Make sure RACF is properly installed using the service and install procedures as discussed in *Program Directory for RACF Security Server for z/VM*.

Troubleshooting Table: Documentation

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, consider using the Readers' Comment form at the back of that book to report the documentation error. You have your choice of mailing in a hardcopy comment sheet (the "Readers' Comments" form) or sending us the comment by FAX or electronic mail (see the "Communicating Your Comments to IBM" page in the back). Report the problem to the IBM support center for the following 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 34 b. Chapter 3, "Searching for a Matching Problem," on page 43 c. Chapter 4, "Reporting RACF Problems to IBM," on page 45.

Troubleshooting Table: Incorrect Output

This section 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. Message text could include invalid 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. Check that error message handling is ON. This can be done with the command SET EMSG ON (not SET EMSG TEXT). If a batch job encountered the error, check that the batch console was spooled back to the originating user. 2. Check the global access checking table for an entry describing the resource. 3. Check for an installation exit for the RACHECK macro. 4. Check the kind of RACF 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.) 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 34. 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. 7. Get a printed or online copy of the incorrect output. You will need the copy when you analyze the problem or report the problem to IBM. 8. See Chapter 3, “Searching for a Matching Problem,” on page 43. 9. See Chapter 4, “Reporting RACF Problems to IBM,” on page 45.

Table 5. Troubleshooting Table: Incorrect Output (continued)

Symptom	Recommended Procedures for Incorrect Output
Incorrect output: Messages	**Action:** 1. Make sure 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 batch console was spooled back to the originating user. If an interactive user had the problem, check that the user had SET EMSG ON (not SET EMSG TEXT). 2. Decide which of the following 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 "Troubleshooting Table: Messages" on page 30. b. Was the message itself invalid (for example, did it contain invalid or unreadable data, or was an error message issued for a nonerror situation?) 4. If you had incorrect output from a RACF macro, see "Common Usage Problems with RACF Macros and Utilities" on page 5 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 34. 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. 6. Get a printed or online copy of the incorrect output. You will need the copy when you 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 Chapter 3, "Searching for a Matching Problem," on page 43. 8. See Chapter 4, "Reporting RACF Problems to IBM," on page 45.

Table 5. Troubleshooting Table: Incorrect Output (continued)

Symptom	Recommended Procedures for Incorrect Output
Incorrect output: Invalid 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. Check that error message handling is ON. This can be done with the command SET EMSG ON (not SET EMSG TEXT). If a batch job encountered the error, check that the batch console was spooled back to the originating user. 2. Decide what the nature of the abend error is. a. Was the abend valid but unexpected? Go to "Troubleshooting Table: Abends" on page 10. b. Was the abend not documented? Go to "Troubleshooting Table: Documentation" on page 21. c. Was the abend itself invalid? For example, was the abend issued for a nonerror situation? d. If you had incorrect output from a RACF macro, see "Common Usage Problems with RACF Macros and Utilities" on page 5 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 34. 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. 4. Get a printed or online copy of the incorrect output. You will need the copy when you analyze the problem or report the problem to IBM. Get dump output and dump analysis results (see box below). **To Get the Dump Output and Analysis Results** Use the following procedures: a. "Obtaining a System Dump on z/VM" on page 39, if you need to request a dump b. "Formatting a System Dump on z/VM" on page 40 c. "Validating Input" on page 41. 5. See Chapter 3, "Searching for a Matching Problem," on page 43. 6. See Chapter 4, "Reporting RACF Problems to IBM," on page 45.

Table 5. Troubleshooting Table: Incorrect Output (continued)

Symptom	Recommended Procedures for Incorrect Output

Incorrect output: Improperly formatted output from RACF LIST commands (LISTDSD, LISTUSER, and RLIST)	**Action:**
	1. Check whether any error messages were issued for the user request and make sure that message IDs are included with the messages.
	Check that error message handling is ON. This can be done with the command SET EMSG ON (not SET EMSG TEXT). If a batch job encountered the error, check that the batch console was spooled back to the originating user.
	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 "RACF database" on page 26.
	• RACF's processing of the command request.
	3. See "Performing a Preliminary Search for a Matching Problem" on page 34.
	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.
	4. Get a printed or online copy of the incorrect output. You will need the copy when you analyze the problem or report the problem to IBM.
	Enter:
	`cp spool console start *`
	`  (enter the racf command(s) which cause the incorrect output)`
	`cp spool console close`
	The output from the command will be on the reader list of the user issuing the commands.
	5. See Chapter 3, "Searching for a Matching Problem," on page 43.
	6. See Chapter 4, "Reporting RACF Problems to IBM," on page 45.

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. Check that error message handling is ON. This can be done with the command SET EMSG ON (not SET EMSG TEXT). If a batch job encountered the error, check that the batch console was spooled back to the originating user. 2. See "Performing a Preliminary Search for a Matching Problem" on page 34. 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. 3. Get a printed or online copy of the improperly formatted report. You will need the copy when you 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 do the following: • Print the SMF DATA file using the CMS PRINT command. As a check on DSMON, use the RACF LIST commands to check the profiles that seem to be reported incorrectly by DSMON. 5. See Chapter 3, "Searching for a Matching Problem," on page 43. 6. See Chapter 4, "Reporting RACF Problems to IBM," on page 45.

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. Check that error message handling is ON. This can be done with the command SET EMSG ON (not SET EMSG TEXT). If a batch job encountered the error, check that the batch console was spooled back to the originating user. 2. If the RACF report writer leads you to suspect that the SMF (system management facilities) records are invalid, check to make sure that the SMF records are at fault, not the RACF report writer. 3. See "Performing a Preliminary Search for a Matching Problem" on page 34. 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. 4. Get a printed or online copy of the incorrect output. You will need the copy when you analyze the problem or report the problem to IBM. For SMF records (types 80 or 81): Use the RACF report writer to list the contents of the RACF SMF records. (See *z/VM: RACF Security Server Auditor's Guide*.) As a check on the RACF report writer, you can do the following: • Print the SMF DATA file using the CMS PRINT command with HEX specified. 5. See Chapter 3, "Searching for a Matching Problem," on page 43. 6. See Chapter 4, "Reporting RACF Problems to IBM," on page 45.
Incorrect output: The RACF database (this includes profiles, index blocks, and index 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. Check that error message handling is ON. This can be done with the command SET EMSG ON (not SET EMSG TEXT). If a batch job encountered the error, check that the batch console was spooled back to the originating user. 2. The following RACF functions affect the RACF database: • RACF commands that add, alter, or delete profiles (such as ADDUSER, ALTUSER, DELUSER, RDEFINE, RALTER, and RDELETE). • RVARY and SETROPTS commands. • RACF utilities: BLKUPD and IRRUT400. • RACF macros: RACDEF, RACINIT, ICHEINTY, ICHETEST, ICHEACTN, and RACXTRT (with ENCRYPT or TYPE=REPLACE specified). 3. If you had incorrect output from a RACF macro, see "Common Usage Problems with RACF Macros and Utilities" on page 5 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 34. 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. 5. Get a printed or online copy of the incorrect output. You will need the copy when you analyze the problem or report the problem to IBM. Run the IRRUT200 and BLKUPD utilities to get a description of the RACF database. See *z/VM: RACF Security Server System Programmer's Guide* for a description of how to use these utilities. 6. See Chapter 3, "Searching for a Matching Problem," on page 43. 7. See Chapter 4, "Reporting RACF Problems to IBM," on page 45.

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. Check that error message handling is ON. This can be done with the command SET EMSG ON (not SET EMSG TEXT). If a batch job encountered the error, check that the batch console was spooled back to the originating user. 2. Decide if the return or reason code is: • Invalid • Undocumented. See "Troubleshooting Table: Documentation" on page 21. 3. See "Common Usage Problems with RACF Macros and Utilities" on page 5 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 34. 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. 5. Get a printed or online copy of the incorrect output. You will need the copy when you analyze the problem or report the problem to IBM. Get dump output and dump analysis results (see box below). If the macro return or reason code was incorrect, request a system dump when RACF returns control from a RACF macro request that produced the incorrect output. **To Get the System Dump Output and Analysis Results** Use the following procedures: a. "Obtaining a System Dump on z/VM" on page 39, if you need to request a dump b. "Formatting a System Dump on z/VM" on page 40 c. "Validating Input" on page 41. 6. See Chapter 3, "Searching for a Matching Problem," on page 43. 7. See Chapter 4, "Reporting RACF Problems to IBM," on page 45.
Incorrect output: Return codes from the 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. Check that error message handling is ON. This can be done with the command SET EMSG ON (not SET EMSG TEXT). If a batch job encountered the error, check that the batch console was spooled back to the originating user. 2. Decide if the return code is: • Invalid • Undocumented. See "Troubleshooting Table: Documentation" on page 21. 3. See "Performing a Preliminary Search for a Matching Problem" on page 34. 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. 4. Get a printed or online copy of the invocation of the RACF report writer and the return code output. You will need the copy when you analyze the problem or report the problem to IBM. 5. See Chapter 3, "Searching for a Matching Problem," on page 43. 6. See Chapter 4, "Reporting RACF Problems to IBM," on page 45.

Troubleshooting Table: Logon Problems

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. Check that error message handling is ON. This can be done with the command SET EMSG ON (not SET EMSG TEXT). If a batch job encountered the error, check that the batch console was spooled back to the originating user. 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. 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 profile. See *z/VM: RACF Security Server Security Administrator's Guide* for more information on the REVOKE attribute. 4. See "Performing a Preliminary Search for a Matching Problem" on page 34. 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. 5. Get a printed or online copy of the SMF records related to the logon problem. You will need the copy when you analyze the problem or report the problem to IBM. 6. See Chapter 3, "Searching for a Matching Problem," on page 43. 7. See Chapter 4, "Reporting RACF Problems to IBM," on page 45.

Troubleshooting Table: Loops

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. Check that error message handling is ON. This can be done with the command SET EMSG ON (not SET EMSG TEXT). If a batch job encountered the error, check that the batch console was spooled back to the originating user. 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 your system diagnosis guide 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 the following 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 34. c. If you did not find a matching problem, you will need to request a dump (if you don’t have one already) and collect problem information to report to IBM. Use the following procedures: 1) “Obtaining a System Dump on z/VM” on page 39, if you need to request a dump 2) “Formatting a System Dump on z/VM” on page 40 3) “Validating Input” on page 41. d. See Chapter 3, “Searching for a Matching Problem,” on page 43 e. See Chapter 4, “Reporting RACF Problems to IBM,” on page 45.

Troubleshooting Table: Messages

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 TSO parsing services which are supplied in the RACF Security Server for z/VM. The RAC command processor also issues these messages; you may see these messages on z/VM during a RACF command session or when issuing commands with the RAC command processor. **Action:** 1. See *z/VM: RACF Security Server 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 manual. 2. If you cannot correct the problem from the information in the messages manual, do the problem determination procedure related to the message.
Messages with a prefix of ICH or IRR.	**Description:** These are RACF messages that you may see on either an z/OS or a z/VM system. **Action:** 1. See *z/VM: RACF Security Server Messages and Codes* for a description of the message and the condition causing it. Attempt to correct the problem from the information in *z/VM: RACF Security Server Messages and Codes*. 2. If you cannot correct the problem from the information in *z/VM: RACF Security Server Messages and Codes*, do the problem determination procedure related to the message.
Messages with a prefix of CST or RPI	**Description:** CST and RPI messages are issued by the RACF service machine. **Action:** 1. See *z/VM: RACF Security Server Messages and Codes* for a description of the message and the condition causing it. Attempt to correct the problem from the information in *z/VM: RACF Security Server Messages and Codes*. 2. If you cannot correct the problem from the information in *z/VM: RACF Security Server Messages and Codes*, do the problem determination procedure related to the message.
Messages with a prefix of HCPRPI, HCPRPD, or HCPRPW.	**Description:** Messages prefixed with HCPRPI, HCPRPD, or HCPRPW are RACF for z/VM messages. **Action:** 1. See *z/VM: RACF Security Server Messages and Codes* for a description of the message and the condition causing it. Attempt to correct the problem from the information in *z/VM: RACF Security Server Messages and Codes*. 2. If you cannot correct the problem from the information in *z/VM: RACF Security Server Messages and Codes*, do the problem determination procedure related to the message.
Messages with a prefix of HCP, but not HCPRPI, HCPRPD, or HCPRPW.	**Description:** These are CP 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 on correcting this problem.
Messages with a prefix of DMS	**Description:** Messages prefixed with DMS and issued from the RACF service machine are RACF for z/VM messages. **Action:** For more information, use *System Messages and Codes* for your release of CMS.

Table 8. Troubleshooting Table: Messages (continued)

Symptom	Recommended Procedures for Messages
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 user or batch job had SET EMSG ON (not SET EMSG TEXT). 2. See *z/VM: RACF Security Server 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 manual. 3. If the message is not in *z/VM: RACF Security Server Messages and Codes*, and you are certain it is a RACF message, see “Troubleshooting Table: Documentation” on page 21.

Troubleshooting Table: Performance Problems

Table 9. Troubleshooting Table: Return Codes

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.	**Action:** 1. Most performance problems related to RACF are corrected by changing the way RACF is set up. Before suspecting that RACF is causing performance problems, review the following items to assure that the problem is not caused by your installation's RACF setup. **Pay close attention to recent changes made in the way you have set up RACF**. For more information, see *z/VM: RACF Security Server System Programmer's Guide*. __ a. The rate at which users log on affects system performance. The higher the rate, the lower the performance. You might see peaks and valleys in performance through the day. __ b. Using the RACLIST or GENLIST operands on the SETROPTS command can improve performance. __ c. Use of the global minidisk table can improve performance. __ d. Use of the global access checking table can improve performance. __ e. The processing performed by installation-written exit routines can affect performance. __ f. The placement of the RACF database on DASD (including how many DASDs the database is placed on) can affect performance. __ g. The use of logging by users with SPECIAL or group-SPECIAL attributes or users with AUDITOR or group-AUDITOR attributes can reduce performance. __ h. The gathering of resource statistics can reduce performance. __ i. The use of certain RACF commands (including SEARCH, LISTDSD with the ID or PREFIX operands, LISTGRP *) can reduce performance when there is a single RACF database. __ j. The use of RACF utilities can reduce performance when there is a single RACF database. __ k. The use of resident data blocks can improve performance. __ l. During failsoft processing, the operator is prompted frequently, degrading performance while you repair RACF. To improve performance during failsoft processing, write a RACHECK preprocessing exit. __ m. List of groups processing can reduce performance. __ n. Group authority processing when made to percolate for a user to subgroups owned by the group can reduce performance. __ o. Controlling the TRANSFER and TAG commands and the MDISK system function can reduce system performance. See *z/VM: RACF Security Server Security Administrator's Guide* for information on using a z/VM event profile to stop controlling these commands. __ p. Defining too many generic profiles within a general resource class can hurt performance. 2. See "Performing a Preliminary Search for a Matching Problem" on page 34. 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. 3. See Chapter 3, "Searching for a Matching Problem," on page 43. 4. See Chapter 4, "Reporting RACF Problems to IBM," on page 45.

Troubleshooting Table: Return Codes from RACF Macros

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. 2. See "Common Usage Problems with RACF Macros and Utilities" on page 5 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/VM: RACF Security Server Macros and Interfaces*, except for the RACROUTE macro, which is described in *z/VM: 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. Check that error message handling is on. This can be done with the command SET EMSG ON (not SET EMSG TEXT). If a batch job encountered the error, check that the batch console was spooled back to the originating user. 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 book for the program of which the module is a part. To determine which diagnosis or logic book contains information about the program, find the name of the component that owns the module. If the component is owned by IBM, the general information manual for the product containing the component has the name and order number of any diagnosis or logic publication. 6. If the problem still appears to be caused by RACF, use the following diagnostic procedures: a. "Performing a Preliminary Search for a Matching Problem" on page 34. 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 z/VM" on page 39. c. "Formatting a System Dump on z/VM" on page 40. d. "Validating Input" on page 41. e. Chapter 3, "Searching for a Matching Problem," on page 43. f. Chapter 4, "Reporting RACF Problems to IBM," on page 45.
RACF manager return codes	**Action:** 1. See *z/VM: RACF Security Server 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 manual. 2. If you cannot correct the problem from the information in the messages manual, perform the problem determination procedure that is related to the message.

Troubleshooting Table: Waits

Table 11. Troubleshooting Table: Waits

Symptom	Recommended Procedures for Waits
Waits	**Description:** RACF may fall into a wait state for a number of reasons (such as hung I/O, missing attention, or external interrupts). **Action:** 1. First, decide which kind of wait it is: • CP wait • A hang (enabled wait) in a service machine (RACF is not active; requests are not honored). If it is a CP wait, it is likely that CP caused the problem. See your system diagnosis guide. If it is a wait in the RACF service machine, continue with the steps in the remainder of this list. 2. Check to see why CP is not dispatching the service machine. 3. See “Analyzing a Wait Problem on z/VM” on page 42. 4. See “Performing a Preliminary Search for a Matching Problem.” 5. Gather more information: • If you don’t have a z/VM dump of the RACF service machine, see “Obtaining a System Dump on z/VM” on page 39 and “Formatting a System Dump on z/VM” on page 40. 6. See Chapter 3, “Searching for a Matching Problem,” on page 43. 7. See Chapter 4, “Reporting RACF Problems to IBM,” on page 45.

Performing a Preliminary Search for a Matching Problem

Prerequisites

“Troubleshooting Tables for Problems in RACF” on page 9 has steps to verify that this is a problem caused by RACF and to collect some initial problem symptoms.

Introduction to Procedure

This topic tells you 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 the following for a matching problem (which could have a fix 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, in order 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.

Procedure

In this procedure, you perform the following steps:

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

Details of the procedure follow:

1. Use Table 12 on page 36 to select which search arguments apply to the type of problem you have.
2. Using Table 13 on page 37 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 Chapter 3, "Searching for a Matching Problem," on page 43. 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. (Return to where you were in the troubleshooting tables.)
6. If your search has too many matches, consider using the following 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: Chapter 3, "Searching for a Matching Problem," on page 43 summarizes the additional symptoms that you can obtain through in-depth analysis.

Table 12. Preliminary Search Symptoms for RACF in z/VM

Search Argument	Free-Format Example	Structured Example
Component identification (ID) The component ID will match problems that had RACF as either the cause of the problem or the apparent cause (for a problem that was reassigned to another component).	RACFVM or RACF/z/VM 576700201	PIDS/576700201
Release level of RACF. Use the FMID (function modification identifier). For RACF 5.3, the FMID is RACF530.	530	LVLS/530
RACF request that failed	RACHECK PROFILE	PCSS/RACHECK PCSS/PROFILE
Error reason code from the RACF request (if available)	rc00000004 or rc04 or rc4	PRCS/00000004
Associated error message ID The console log might contain error messages associated with the problem that you encountered. Some problems produce multiple error messages, all or some of which may be relevant search arguments. If you have multiple error messages, you should search on various combinations of these messages.	MSCSTABE001T or MSRPIMGR051T or MSGICH408I	MS/CSTABE001T or MS/RPIMGR051T or MS/ICH408I
Associated reason code from message	rc04 or rc4	PRCS/04
RACF module	RPIMJR	RIDS/RPIMJR
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 CP 568411202 or DirMaint™ 5749DVH00. 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).	568411202	PIDS/568411202
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.	UV50605	PTFS/UV50605
Area within a document, such as a part, section, chapter, appendix, or page	Chapter3 appendixA	PUBS/CHAP3 PUBS/APPXA

Table 12. Preliminary Search Symptoms for RACF in z/VM (continued)

Search Argument	Free-Format Example	Structured Example
Document order number Document order numbers end with a version number, such as GC28-1366-08. All version numbers are now two digits, although previously they could be one. You could leave off the second hyphen and the level numbers that follow it. Then, if you receive too many matches, add the version numbers to the search symptom.	GC28-1366-08	PUBS/GC28136608

Examples of Search Arguments for RACF on z/VM

Table 13 has examples of free-format search arguments for problems in RACF (z/VM).

Table 13. Examples of Search Arguments for RACF on z/VM

Problem Type	Example of Search Argument	Descriptions of Search Symptoms
Abend	576700201 abend685 rc04	Component ID Abend code Abend reason code (if one exists)
Documentation error	576700201 doc RACHECK	Component ID The word *doc* The name of the RACF request with the documentation error
Incorrect output	576700201 RACHECK output	Component ID The type of output request that was involved The word *output*
Loop	576700201 RPIMJR 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	576700201 MSCSTABE001T invalid password	Component ID The message identifier Any inserted code (or text) in the message
Performance degradation	576700201 RACHECK performance	Component ID The name of any associated RACF request The word *performance*
Return code from macro	576700201 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)	576700201 RPIMJR enabled wait or 576700201 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).

Chapter 2. Collecting and Analyzing Problem Data

This chapter explains how to collect and analyze problem data.

Collecting Problem Data

If you don't know what data to collect for the problem that you are diagnosing, see "Troubleshooting the Problem" on page 4. Also, Chapter 4, "Reporting RACF Problems to IBM," on page 45 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 section on conducting an audit in the *z/VM: RACF Security Server Auditor's Guide*.

Obtaining a System Dump on z/VM

Prerequisites

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

Introduction to Procedure

This procedure sets a TRACE to intercept program checks that occur. The TRACE set by this procedure causes a VMDUMP to be generated before the system takes any recovery action. You may want to re-create the problem you are investigating (or set up the conditions to allow the problem to reoccur) after setting the trap.

Procedure:

1. Log on to the RACF service machine that experienced the original problem (after RACF is reinitialized).
2. Use the CP DISPLAY command to display the storage contents of the 4 bytes at location X'6C':

   ```
   DISPLAY 6C.4
   ```

   Sample results follow:

   ```
   E6C058
   ```
3. Write down the address shown. In this case, the address is `E6C058`.

4. Set a TRACE for that address, as follows:

   ```
   CP TR I PSWA E6C058.4 CMD VMDUMP
   ```

   This command requests a VMDUMP to be generated when control arrives at location X'E6C058'.

Checking the Trap

To check the trap, query the TRACE as follows:

```
Q TR
```

The following will result:

```
INSTR   PSWA  00E6C058-00E6C05B
TERM    NOPRINT  NORUN  SIM
SKIP 00000  PASS 00000  STOP 00000  STEP 00000
CMD  VMDUMP
```

Formatting a System Dump on z/VM

Prerequisites

You may need to set a PER trap to obtain the system dump.

Introduction to Procedure

RACF-specific information is not available in system dumps generated on z/VM.

Procedure

To debug a z/VM dump generated for a RACF service machine, use standard debugging methods appropriate for debugging z/VM dumps.

Note: Make sure to have a current CMS load map for the RACF service machine that experienced the problem. The CMS that runs in a RACF service machine has been enhanced for RACF function.

Consider studying the following data for problem information:

- The symptom record
- The NUCON data area
- General-purpose registers.

Analyzing Console Logs

Prerequisites

Error messages must be issued for the RACF request involved in the problem and the console must be spooled back to the originating user.

Check that the user (or the RACF service machine) had SET EMSG ON (not SET EMSG TEXT).

Introduction to Procedure

This procedure collects logs from the operator console or a secondary console.

Messages issued to the system operator or to a secondary console could be related to a RACF problem.

RACF messages issued as a result of user requests can help determine the sequence of events. These messages are sent to the terminal of an interactive user and to the primary RACF service machine (usually RACFVM) when RACF macros are invoked.

Procedure

For problems involving RACF macros, check the console log for the RACF service machine.

Validating Input

This section explains how to validate input to RACF processes.

Checking the Parameter List Passed to RACF Macros on z/VM

Prerequisites

Call the IBM support center to get the virtual instruction address at which an RACF routine receives control when invoked.

Introduction to Procedure

This procedure allows you to stop execution of the RACF service machine at a virtual instruction address.

Procedure:

1. Enter the following command:

   ```
   CP TRACE INSTR FROM hexloc
   ```

   where *hexloc* is the hexadecimal location (address) of an instruction in an IBM module supplied by the IBM support center.
2. Do one of the following:
   - If the RACROUTE macro was issued, display general register zero, which contains the address of the parameter list passed to the RACF routine.

     ```
     CP DISPLAY GO
     ```

     Write down the address displayed.
   - If another RACF macro was issued, display general register 1, which contains the address of the parameter list passed to the RACF routine.

     ```
     CP DISPLAY G1
     ```

     Write down the address displayed.
3. Get the length of the input parameter list from one of the following:
   - For parameter lists for installation exits, see *z/VM: RACF Security Server Diagnosis Guide*.
   - For parameter lists for RACF macros, see Chapter 6, “Parameter Lists,” on page 87.
4. Enter the following command:

   ```
   CP DISPLAY Taddress.length
   ```

   where *address* is the address from the previous step and *length* is the length of the parameter list.

Analyzing a Wait Problem on z/VM

1. A good place to start is with the PSW (program status word) and general regulations. Look to see whether RACF is enabled or disabled, and where in the code RACF is waiting. Is RACF waiting for I/O to complete?

   It is important as well to examine low-core fixed storage locations for PSWs, especially EXT old and new PSWs as well as CSW (control status word) status. It is also beneficial to examine certain flags in the RACF service machine's VMDBK:

   ```
   VMDCFLAG  (VMDBK offset X'381')
   VMDOSTAT  (VMDBK offset X'382')
   VMDSTAT2  (VMDBK offset X'383')
   VMDRSTAT  (VMDBK offset X'509')
   ```

   For descriptions of the values within these flags, see the VMDBK data area in *z/VM: CP Data Areas and Control Blocks*.

2. If everything still looks good at this point, then examine the processing that occurs within RACF. Call the IBM support center for assistance.

Chapter 3. Searching for a Matching Problem

Prerequisites:

1. See “Performing a Preliminary Search for a Matching Problem” on page 34 for a description of easily located problem symptoms to use to find a matching problem.
2. See “Troubleshooting the Problem” on page 4 for the diagnostic procedures to follow to collect additional symptoms needed for a search.

   For problems involving system dumps, you will need the following diagnostic procedures:

   - “Obtaining a System Dump on z/VM” on page 39 and “Formatting a System Dump on z/VM” on page 40.

Introduction to Procedure

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 34.

Procedure

Review Table 14 for other RACF symptoms that you might want to use in your search for a matching problem (or to report to IBM). The table lists some of the most probable search symptoms for problems in RACF.

Note: For abends, additional symptoms such as a register/PSW (program status word) difference and a failing instruction area are automatically produced in DAEDATA output; they are not repeated here. Notes about the symptoms are indicated, where applicable.

Table 14. Additional Search Symptoms for RACF

Description of Search Symptom	Free-Format Example	Structured Example
Control block name	ACEE	FLDS/ACEE
Control block field name (see note 1)	ACEEFLG1	FLDS/ACEEFLG1
Symptoms for later searches		
Level of failing module (see note 2 on page 44)	HRF1802 or UY09958	LVLS/HRF1802 or PTFS/UY09958

Notes:

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.

Chapter 4. Reporting RACF Problems to IBM

Prerequisites

"Troubleshooting the Problem" on page 4 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, which are explained in *Programming System General Information Manual (PSGIM)*.

Introduction to Procedure

This section describes the information that you will 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 will need to describe your system and the problem that you experienced. The people at the IBM support center will 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 34 and Chapter 3, "Searching for a Matching Problem," on page 43 have advice on which RACF symptoms to use in search arguments for search tools.

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

Procedure

Before calling IBM, collect the information called for in the worksheet for your type of problem. See Table 15 on page 46 through Table 16 on page 54 for the applicable worksheet.

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

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

Recommendation:

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.

For the complete list of diagnostic procedures for each type of problem, see "Troubleshooting Tables for Problems in RACF" on page 9.

Table 15. Information to Gather before Calling IBM with Abend Problems

Information to Gather before Calling IBM with Abend Problems

You can use this worksheet as a guide for making your own notes elsewhere.

1. Level of RACF: _____

(Example: RACF Security Server for z/VM 5.3.0)

2. Level of operating system: _____

3. Level of Service: _____

(Example: RSU tape 9603)

4. APAR and/or PTF level of abending module: _____

(Example: VM57305)

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, RACF options changed, new program products, new hardware):

8. The message that contains the abend code: _____ Also, **complete** message text for:

CSTABE001T, CSTDET001I, CSTINT005I, or CSTPGM00xT:

IRR401I or ICH409I:

9. Assembler listing, if source is available.

10. If a hardware problem is suspected, EREP (error recording and editing program) records.

Information to Gather before Calling IBM with Documentation Problems

You can use this worksheet as a guide for making your own notes elsewhere.

1. Level of RACF: _____

(Example: RACF Security Server for z/VM 5.3.0)

2. Level of operating system: _____

3. Level of Service: _____

(Example: RSU tape 9603)

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. Chapter and Topic Heading:

8. Page or Topic number: _____

Information to Gather before Calling IBM with Output Problems

You can use this worksheet as a guide for making your own notes elsewhere.

1. Level of RACF: _____

(Example: RACF Security Server for z/VM 5.3.0)

2. Level of operating system: _____

3. Level of Service: _____

(Example: RSU tape 9603)

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:

(Example: Publication title, order number, and page or topic number)

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.

Information to Gather before Calling IBM with Logon Problems

You can use this worksheet 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 RACF: _____

(Example: RACF Security Server for z/VM 5.3.0)

2. Level of operating system: _____

3. Level of Service: _____

(Example: RSU tape 9603)

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.

Information to Gather before Calling IBM with Loop Problems

You can use this worksheet as a guide for making your own notes elsewhere.

1. Level of RACF: _____

(Example: RACF Security Server for z/VM 5.3.0)

2. Level of operating system: _____

3. Level of Service: _____

(Example: RSU tape 9603)

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:

8. What conditions the loop occurs in:

9. Whether loop is ___ enabled or ___ disabled.

10. If loop is enabled, please include a stand-alone system dump.

11. If applicable, please have available online or (preferably) printed output.

Information to Gather before Calling IBM with Message Problems

You can use this worksheet as a guide for making your own notes elsewhere.

You can use this worksheet as a guide for making your own notes elsewhere.

1. Level of RACF: _____

   (Example: RACF Security Server for z/VM 5.3.0)

2. Level of operating system: _____

3. Level of Service: _____

   (Example: RSU tape 9603)

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:

Information to Gather before Calling IBM with Performance Problems

You can use this worksheet as a guide for making your own notes elsewhere.

1. Level of RACF: _____

   (Example: RACF Security Server for z/VM 5.3.0)

2. Level of operating system: _____

3. Level of Service: _____

   (Example: RSU tape 9603)

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).

Information to Gather before Calling IBM with Return Code Problems

You can use this worksheet as a guide for making your own notes elsewhere.

1. Level of RACF: _____

(Example: RACF Security Server for z/VM 5.3.0)

2. Level of operating system: _____

3. Level of Service: _____

(Example: RSU tape 9603)

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 or topic number of your copy of the document in which you found the description of the return code:

Order number: _____ Page or Topic number: _____

10. Assembler listing of program.

Table 16. Information to Gather before Calling IBM with Wait Problems

Information to Gather before Calling IBM with Wait Problems

You can use this worksheet as a guide for making your own notes elsewhere.

1. Level of RACF: _____

(Example: RACF Security Server for z/VM 5.3.0)

2. Level of operating system: _____

3. Level of Service: _____

(Example: RSU tape 9603)

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.

Part 2. Diagnosis Reference for RACF

This part of the document contains additional diagnosis reference material, including troubleshooting information, parameter lists, process flows, a control block overview, and problem determination for a RACF service machine.

Chapter 5. Troubleshooting Your RACF Database

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

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 may uncover problems in your database that you find only when reading the profile. This utility reads every profile.

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

You may 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/VM: RACF Security Server System Programmer's Guide*.

Format of the RACF Database

This section describes the format of the RACF database.

Each RACF database is a non-VSAM single extent data set, made up of 4KB blocks.

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.
- **Profiles (entity records):** contain descriptions of the attributes and authorities for every entity defined to RACF. These entities are:
  - User profiles
  - Group profiles
  - DASD data set profiles
  - Profiles for resources defined by entries in the class descriptor table.

Figure 1 on page 58 illustrates the format of the RACF database.

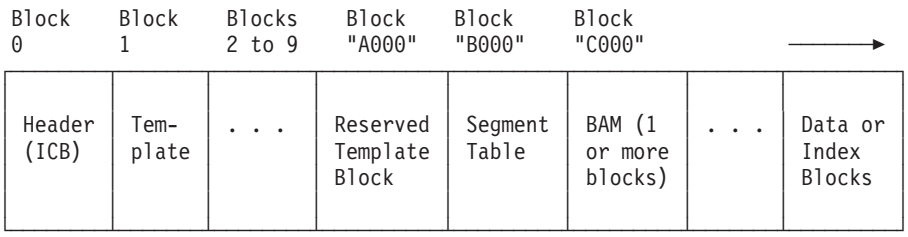


Figure 1. Format of the RACF Database

The first byte in each block is an identifier. Possible 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 “ICB” on page 145 for the fields shown in the data area, ICB.

Templates

IRRTEMP2 contains the database profile mappings. When your installation creates the RACF database, IRRMIN00 reads the mappings, formats them, and writes them to the RACF database. The templates are written to the RACF database, when the RACFCONV EXEC or the RACINITD EXEC executes IRRMIN00.

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 three bytes of the template block will contain 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:

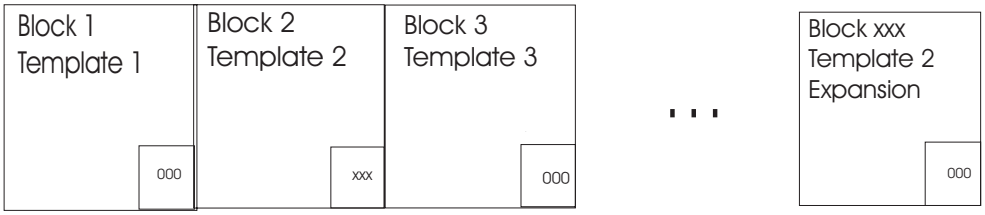


Figure 2. Template blocks

CAUTION:
Customers should not modify the IBM-supplied template definitions, or unpredictable results may occur.

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

Each 4KB BAM (block availability mask) block contains header information followed by block masks. The BAM blocks determine the availability of all the blocks in a RACF database. A RACF database has a block size of 4096. As a result, a BAM block represents up to 2038 blocks of length 4096. The field that shows the space allocation of a data or index block is 16 bits; each bit represents 256 bytes. The database has a bit off, which signals 256 bytes of data space used in a RACF database block.

Header

The header contains the following information:

- **Bytes 1–6:** The RBA of the previous BAM block (or 0 if this is the first block).
- **Bytes 7–12:** The RBA of the next BAM block (or 0 if this is the last block).
- **Bytes 13–18:** The RBA of the first block whose space this BAM block defines.
- **Bytes 19–20:** The number of blocks whose space this BAM block defines. (A maximum of 2038 blocks of length 4096 can be defined by one BAM block.)

Block Masks

Two bytes of the block masks define the status of one RACF database block. Each bit corresponds to a 256-byte slot within a block. If a bit is set to 1, the corresponding slot is free. In Figure 3 on page 60, only the first six slots are in use.

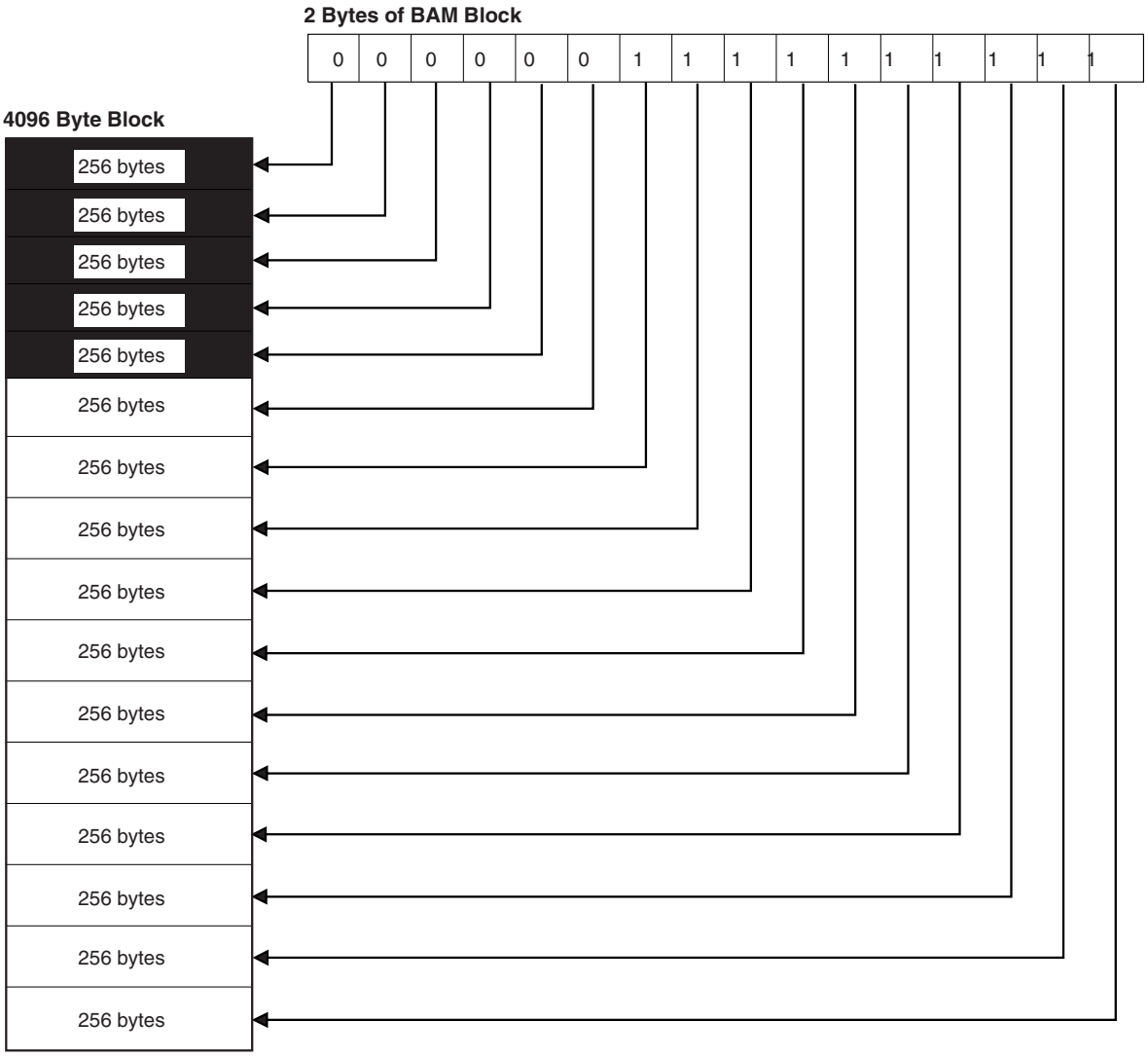


Figure 3. Block Masks of the Database

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.

Index Blocks

RACF uses a multilevel index set to locate profiles. There can be up to 10 levels on a 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.

Figure 4 is an example of part of a RACF index. (This example assumes 3 levels. The level-1 blocks for the first and third level-2 blocks are partially shown in the figure.) The figure illustrates the path RACF uses to find the profile named FAL.N.

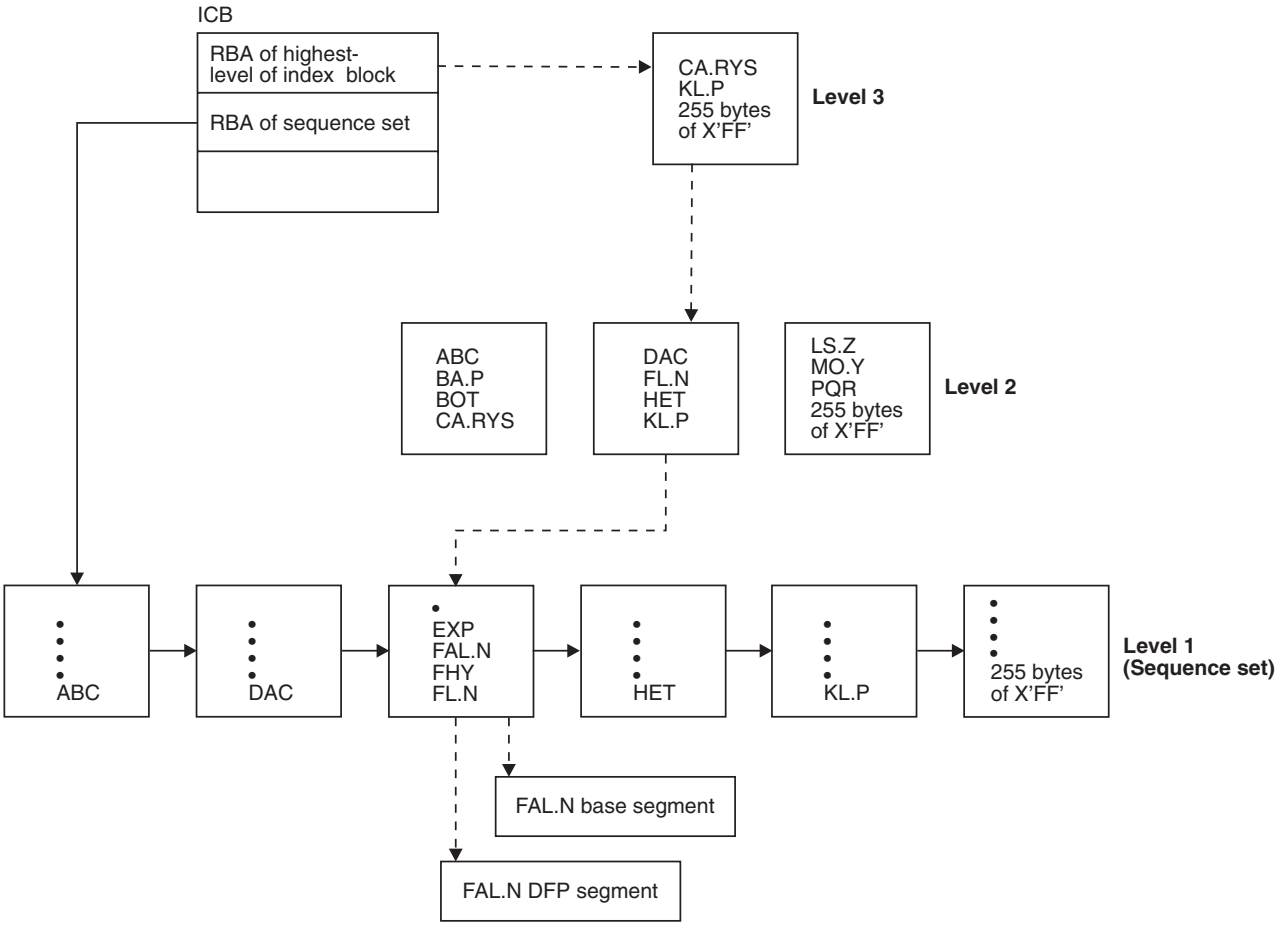


Figure 4. Example of a RACF Index for the Database

Index Block Format on the Database

Each index block contains a 14-byte header followed by the index entries. Level-one index blocks also have sequence set chain pointers, which follow the last entry in the block. At the logical end of each block is a 1-byte end-of-block delimiter, which is set to X'0C'. A level-1 block consisting of 255 bytes of X'FF' signifies the end of the index entries.

Figure 5 on page 62 is an example that gives an overview of the format of an index block. In the example, offset X'0ED4' is the offset to the last entry, entry *n*. The value *n* is the number of entries, and it is also found in the header.

At the end of the block is an offset table. Each offset in the table is 2 bytes. The offset table grows inward, so that the offset for index entry 1 is the very last offset in the table, the offset for index entry 2 is the next-to-the-last offset in the table, and so on.

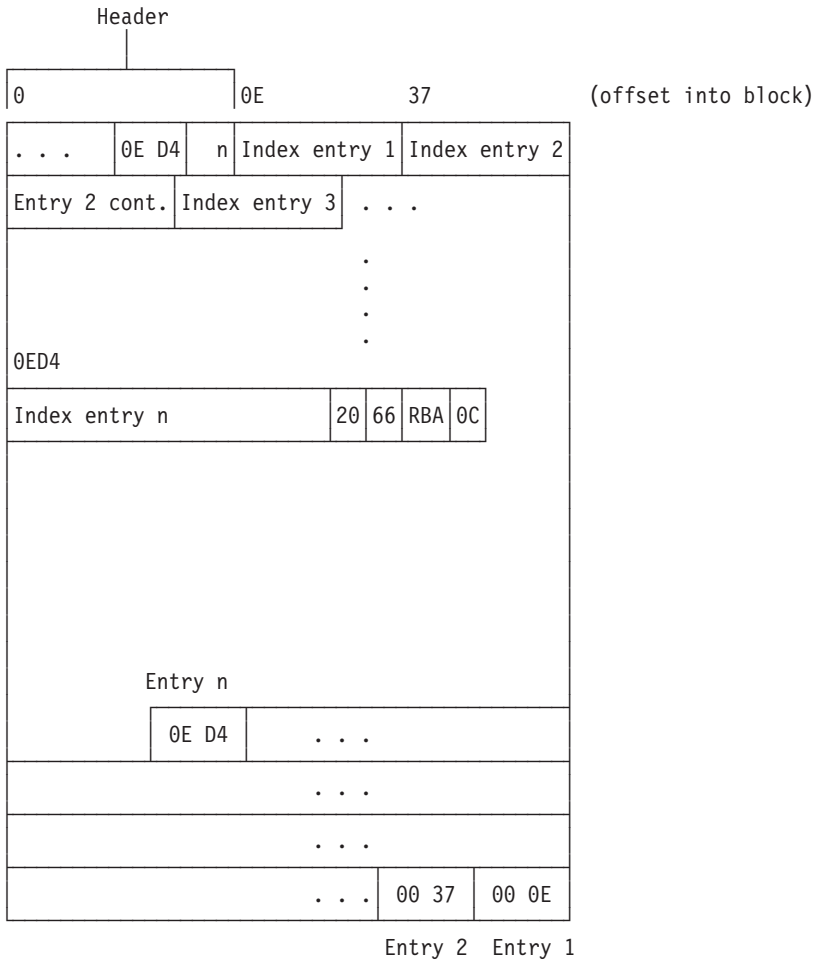


Figure 5. Format of the Index Block

Header

The format of the header is:

Byte 1: The index block identifier (X'8A')
Bytes 2–3: The length of the index block (X'1000')
Byte 4: An index block identifier (X'4E')
Byte 5: Format identifier (X'00')
Byte 6: The index-level number
Bytes 7–8: The offset to the last entry in the index block
Bytes 9–10: The offset to free space in the index block
Bytes 11–12: The offset to a table of index-entry offsets
Bytes 13–14: A count of entries in the table of index-entry offsets.

Figure 6 on page 63 shows an example of the index block header.

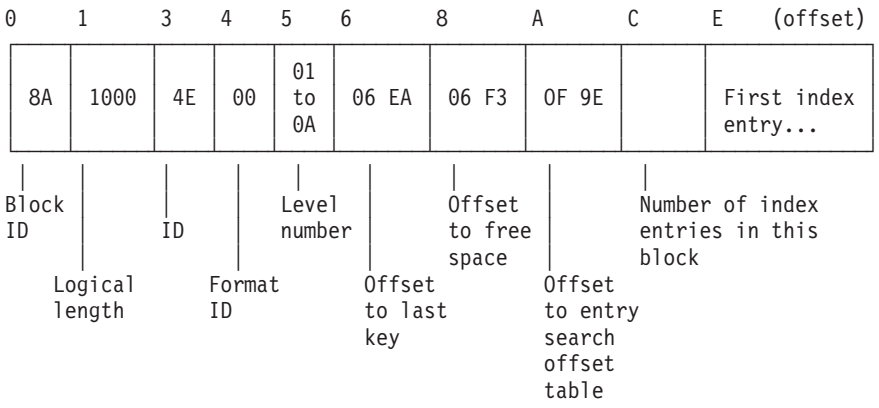


Figure 6. Example of an Index Block Header

Index Entries

The format of each index entry in the block is:

- 1-byte entry identifier, as follows:

  X'20'—pointer entry in level-one block only; identifies the entry containing the pointer to the next sequence set block.

  X'21'—normal index entry (nonduplicate entry), for all index blocks not in level 1, and for the last entry of all index entries in level-1 index blocks that specify the same name.

  X'22'—duplicate entry; indicates the following index entry is a duplicate entry. These exist only for data set profiles.
- 1-byte representation of the type of profile for this entry. The types are:
  - X'01'—Group
  - X'02'—User
  - X'04'—Data set
  - X'05'—General resource
- 2-byte field containing the length of the entry.
- 2-byte entry containing the offset from the beginning of the entry to the segment information.
- 2 bytes containing the front-end compression count.
- 2 bytes containing the length of the profile name.
- Index entry name, 0-255 bytes. Bytes 8-9 contain the actual length.

  An index entry name corresponds to a profile name, but it does not always match letter by letter. It may have been compressed, or some of its characters may have been substituted by RACF for internal purposes (such as to save space or to facilitate the search).

  In an upper-level (that is, not level 1) index block, the index entry name may correspond to the entry name of the last index entry of the next-lower level block. Its rightmost characters may have been 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.
- For non-level-1 blocks:
  - A 1-byte entry pointer identifier (X'62' only); points to lower-level block
  - 6 bytes containing the RBA of the next-level index block

- For level-1 blocks:
  - Segment count—a 1-byte field containing the number of segments for this entry (if there is only a BASE segment, the count will be 1).
  - Segment information (for each segment within the profile):
    - Segment identifier—a 1-byte field identifying a specific segment. See Table 17.
    - Segment RBA—A 6-byte field containing the segment RBA.

Table 17. Segment Identifiers

Group	User	Data Set	General
X'01'—Base	X'01'—Base	X'01'—Base	X'01'—Base
X'02'—DFP	X'02'—DFP	X'02'—DFP	X'02'—Session
	X'03'—TSO		X'03'—DLFDATA (SDOM)
	X'04'—CICS®		
	X'05'—Language		
	X'06'—OPERPARM		
	X'07'—WORKATTR		

Example of a Normal Index Entry

Figure 7 shows an example of a normal index entry, and Figure 8 on page 65 shows the entry data for the entry. The entry is for USR26 with a base, TSO, and DFP segment.

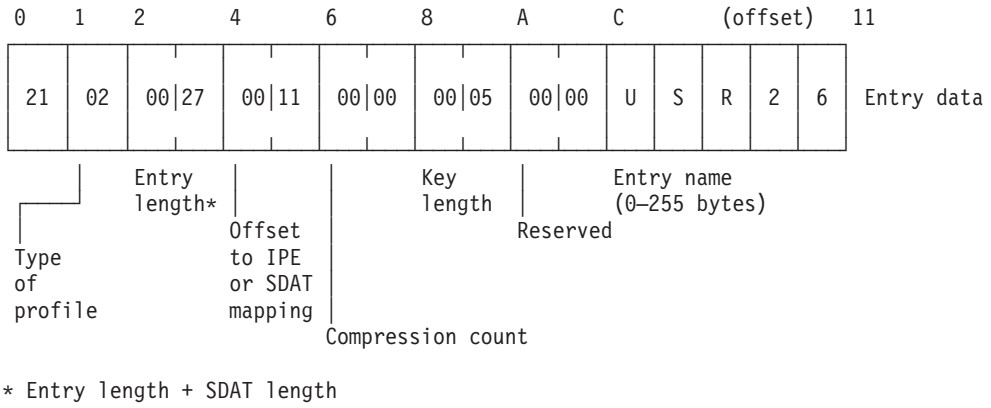


Figure 7. Normal Index Entry for USR26 with a Base, TSO, and DFP Segment

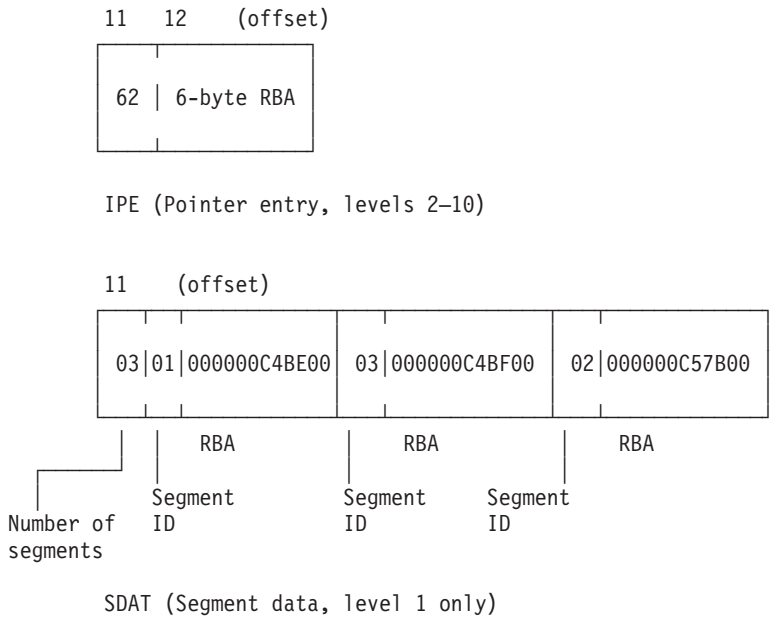


Figure 8. Entry Data for the Normal Index Entry

Example of a Pointer Index Entry

Figure 9 shows a pointer index entry.

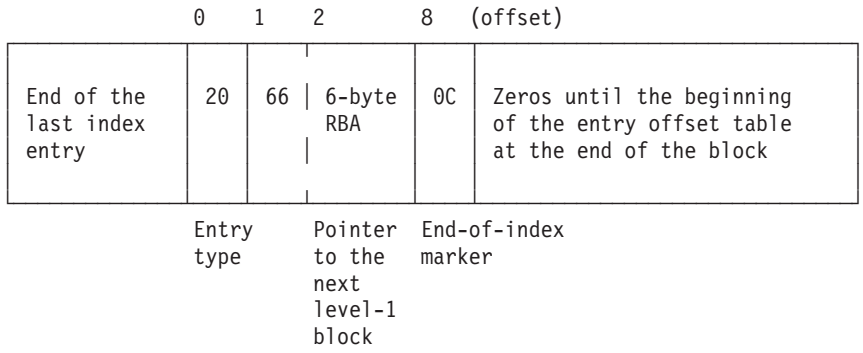


Figure 9. Pointer Index Entry in Level-1 Block

Sequence Set Chain Pointers on the Database

The format of the sequence set chain pointer in each level-1 index block is:

Byte 1: The chain pointer (entry) identifier (X'20')
Byte 2: The entry pointer identifier (X'66')
Bytes 3–8: The RBA of the next level-1 block.

Index-Entry Offset Table at End of an Index Block

At the end of the index block there is a Index Entry Offset Table, which is a table of 2-byte index-entry offsets. RACF uses this information to perform a search on index entries.

Index-Entry Compression

To save space on the RACF database, index entries are compressed. The types of compression are:

- Front-end compression
- Upper-level index name compression.

Front-End Compression

Front-end compression leaves the first entry in an index block unchanged. In all other entries of the block, the compression removes the leftmost characters that are identical to the leftmost characters of the first entry; however, the compression does not necessarily occur when the entry is first added. The index entry will contain a count of the number of characters removed from each entry name.

For example, given the following index block:

```
FEG.ABC
FEG.ADE
FEG.F
FEX.P
FOM
GES.B
```

RACF compresses the entries of the block, preceded by the compression counts, as follows:

```
0 FEG.ABC
5 DE
4 F
2 X.P
1 OM
0 GES.B
```

If an entry in an index block has exactly the same name as the first entry in the block, the entry may be totally compressed.

Upper-Level Index Name Compression

When a name is added to an upper-level (not level-1) index block, RACF compresses the name by eliminating the rightmost characters that are not essential to finding the correct index block at the next-lower level.

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.

Each repeating field on the database is limited to 64KB of customer data.

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/VM: RACF Security Server 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 -`, `VMMDISK -` or `TERMINAL-` for DASD volumes, tape volumes, z/VM minidisks, or terminals, respectively). General-resource class names that are not eight characters in length are padded with trailing blanks.

Generic profile names will 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 will 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 will be 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 may 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 on the pages that follow.

- **Record header**

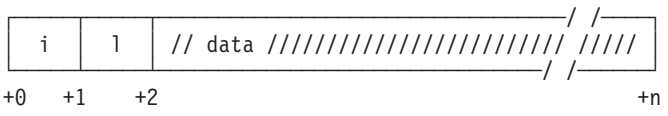
  The record header consists of the following fields:

Field	Description
Record identifier	A 1-byte field containing X'83'
Logical record length	A 4-byte field indicating the portion of the record that actually contains data
Physical record length	A 4-byte field containing the physical record length
Segment name	An 8-byte field containing the name of the segment
Key length	A 2-byte field containing the length of the profile name.
(Reserved)	A 1-byte reserved field
Record key	The profile name
Data fields	One or more segment data fields. Although the potential total length of these fields is 2GB, that may be limited by the type of DASD and the amount of virtual storage available to the user.

- **Field structure**

  All database data fields are in a variable-length format. If a field is not a member of a repeat group, it is stored with a field ID. This field ID is followed by a field indicating the length of the data that follows. If the data is from 1 to 127 bytes, the length field is 1 byte; if the data exceeds 127 bytes (up to 2 to the 31st power) the length field is 4 bytes and has the high-order bit set (for example, X'80000022'). The length field is followed by the data.

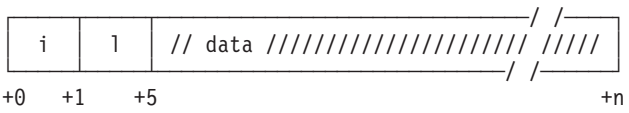
  The following figure shows the structure of a profile field when the data is from 1 to 127 bytes. The structure is:



  i is the field ID. *i* is 1 byte.
  l is the length of the data that follows. *l* is 1 byte.

data is 1 to 127 bytes of data.

The following figure shows the structure of a profile field when the data exceeds 127 bytes. The structure is:



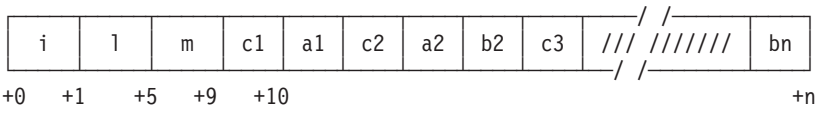
i is the field ID. *i* is 1 byte.
l is the length of the data that follows. *l* is 4 bytes.
data is 128 to 2^{31} bytes of data.

- **Repeat group structure**

  Fields that are part of a repeat group do not have individual ID fields associated with them when they are stored in a profile. They have a field ID associated with the entire repeat group. When RACF writes an occurrence of the repeat group to the RACF database, every field in the occurrence is stored, **even if it contains a null value**.

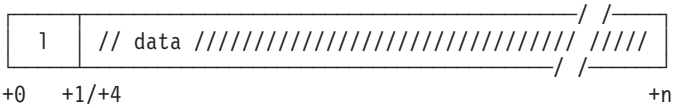
  A count of fields within an occurrence of a repeat group will also be stored with each occurrence.

  The structure of a repeat group is:



  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 the following format:



  l is the length of data that follows. This field is 1 byte if the data is up to 127 bytes and 4 bytes if the data exceeds 127 bytes.

  data is up to 2^{31} bytes of data.

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

Example of a Profile

Figure 10 on page 69 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 the BLKUPD utility to display a profile. See “Block Update Command for a RACF database (BLKUPD)” on page 70.

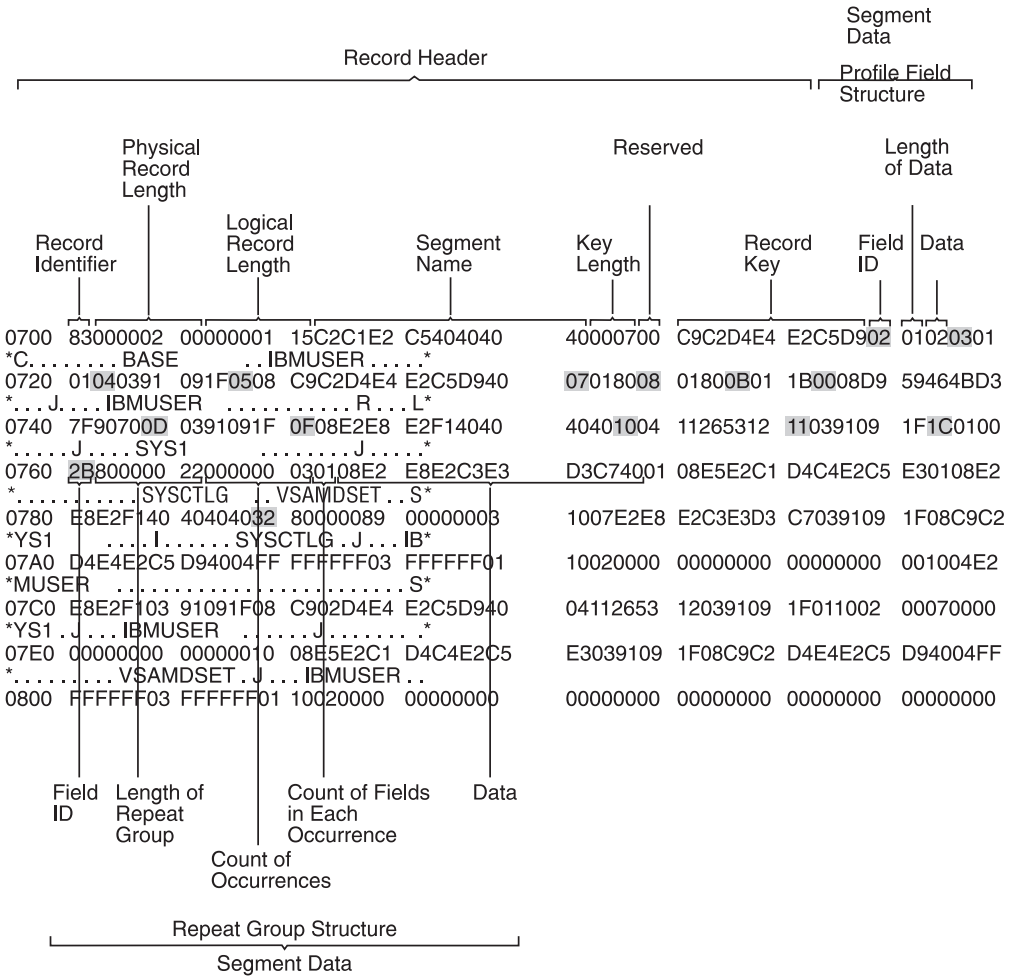


Figure 10. Example of a Profile Displayed by the BLKUPD Utility

Database Unload Utility for a RACF database (IRRDBU00)

If you plan to use the BLKUPD command to update your database, you may 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 may uncover problems in your database that you find only when reading the profile.

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

Block Update Command for a RACF database (BLKUPD)

For the RACF database, the Block Update (BLKUPD) command invokes the BLKUPD utility for processing. Use BLKUPD to examine or modify any BAM, index, or data block on a RACF database.

CAUTION:
Before you use the BLKUPD command, you should be very familiar with the RACF database and its configuration because improper usage of BLKUPD can result in damage to the RACF database. (See "Format of the RACF Database" on page 57.) In general, use this chapter only when directed to by the IBM support center.

You should read and understand the pages on the format of the database before entering the BLKUPD command. Then, before you begin to use the BLKUPD command to perform updates to your RACF database, we recommend your trying to use one of the RACF commands to alter or delete the entry in question.

This section describes the following topics:
- Considerations when using the BLKUPD command
- Hierarchy and uses of the BLKUPD command
- Examples.

Considerations prior to and When Using the BLKUPD Utility

The following considerations apply to the use of the block update utility (BLKUPD):

- In general, you should use this utility only when directed to by the IBM support center.
- Do not use BLKUPD during hours of peak activity on the active RACF database as the command places reserves on the database.
- Before using BLKUPD, try to correct the error by using the RACF commands.
- Before using BLKUPD, make a copy of the RACF database. Use BLKUPD on the copy. IBM may be able to use the copy for further diagnosis.
- Run IRRUT200 before beginning to use BLKUPD. The output may 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.
- Once 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 the RACF command processor to add the profile and index entry.
- Do not use BLKUPD against the BAMs. Use the IRRUT400 utility instead.
- If a RACF database is RACF-protected, you must have at least UPDATE authority to the database.

Hierarchy of the BLKUPD Command

Figure 11 on page 71 shows the relationship between the BLKUPD command and its levels of subcommands.

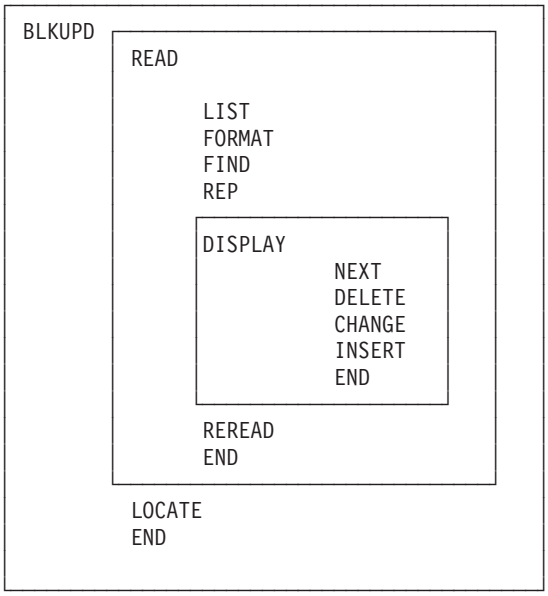


Figure 11. The Relationship between the BLKUPD Command and Its Levels of Subcommands

BLKUPD Command

Use the BLKUPD command to access the RACF database.

Do not use this command during hours of peak activity on the RACF database, because the command places reserves on the database.

The Syntax of the BLKUPD command

```
BLKUPD [racfbname]
```

where *racfbname* specifies the name of a RACF database; *racfbname* can be from 1 to 255 characters.

After BLKUPD is entered, the following subcommands are permitted: LOCATE, READ, or END.

Using the BLKUPD Command

To update or examine the RACF database, perform the following steps:

1. Decide which database you want to work with, and enter BLKUPD.
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.

LOCATE Subcommand on BLKUPD

Use the LOCATE subcommand when you do not know the RBA (relative byte address) of the block you want to address. This subcommand locates an index

entry in the sequence set and prints a formatted list of the contents of the level-1 index block containing the entry name. Optionally, the LOCATE subcommand prints 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]
```

ENTRY(*entryname***)**
specifies the index entry to be located; *entryname* can be from 1 to 255 characters and contain hexadecimal data (for example, `ENTRY(I X'C2' MUSER)`).

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 printed.

The formatted list contains the same information as that produced by the FORMAT subcommand of READ.

If an index block in the chain contains an error, the BLKUPD utility produces a hexadecimal dump of the block. If the name is not found in a level-1 block, this utility prints 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.

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 (just enter another subcommand of the BLKUPD command).

READ Subcommand of BLKUPD

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]
```

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
is specified if 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 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.

NOUPDATE
is specified if 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 the utility, BLKUPD will not allow the changes to be recorded.

Using the Subcommands of the READ Command

The UPDATE keyword on the READ command allows you to use the subcommands to update the block. These subcommands are explained in detail in the following sections.

LIST prints 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 associated with it).

REREAD
overlays the new block in the work area with the old block (valid only with the UPDATE keyword).

END ends the READ function.

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]
```

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-justified and padded with zeros to a byte boundary. Character strings are left justified and padded with zeros to a byte boundary.

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.

LIST Subcommand

The LIST subcommand displays (or prints) 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 the READ

```
LIST [NEW|OLD]
     RANGE(xxx,yyy)
     ALL
```

NEW
specifies the copy of the original block (it may or may 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.

FORMAT Subcommand

The FORMAT subcommand displays (or prints) 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]
```

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.**

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 the READ UPDATE function

```
REP new-string
    OFFSET(xxx)
    [VER(old-string)]
```

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-justified and padded with zeros to a byte boundary. Character strings are left-justified and padded with zeros to a byte boundary.

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.

Note: The verify (VER) operand only verifies the characters specified; it will not verify the whole string. Be sure to enter the entire string you want verified.

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]
```

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 and can contain hexadecimal data.

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 CLASS keyword eliminates the need for prefixing the entry name with the class name.

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.

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 (READ UPDATE)

```
CHANGE [ENTRY(entryname) CLASS(classname)]|[SEGMENT(segment-name)]
       [GENERIC]
       [RBA(rba)]
       [COMP(xx)]
       DUPLICATE|NODUPLICATE
```

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 and may contain hexadecimal data.

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 CLASS keyword eliminates the need for prefixing the entry name with the class name.

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 a discussion of compression and an explanation of index blocks and encoding, see “Index-Entry Compression” on page 65.

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 will be 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 value of *xx* can be from 1 to 8 hexadecimal characters or from 1 to 10 decimal characters.

DUPLICATE
means that you want to make the entry a duplicate of the entry immediately following. This entry that immediately follows may 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 will remain at its previous setting.

Specify only those keywords that you want to change. The others will 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.

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.

The BLKUPD utility displays the new entry; entering the NEXT subcommand displays the original entry again.

Syntax of the INSERT subcommand of DISPLAY (READ UPDATE)

```
INSERT [ENTRY(entryname) CLASS(classname)]|[SEGMENT(segment-name)]
       [GENERIC]
       RBA(rba)
       [COMP(xx)]
       [DUPLICATE|NODUPLICATE]
```

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 and may contain hexadecimal data.

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 CLASS keyword eliminates the need for prefixing the entry name with the class name.

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 a discussion of compression, see “Index-Entry Compression” on page 65.

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 value of *xx* can be from 1 to 8 hexadecimal characters or from 1 to 10 decimal characters.

DUPLICATE
means that you want to make the entry a duplicate of the entry immediately following. This entry that immediately follows may 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.

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.

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 print 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 will 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 in the step above. Continue with the NEXT and CHANGE commands until all the entries in question have been resolved.

It may only be necessary to decompress that second entry; however, based on your examination of the listing, you may have to decompress all the entries. For an explanation of compression, see "Index-Entry Compression" on page 65.

Syntax of the DELETE subcommand of DISPLAY (READ UPDATE)

```
DELETE    [SEGMENT(segmentname)]
D
```

SEGMENT(*segment-name***)**
deletes the specified segment from the current index entry.

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 (READ UPDATE)

```
END     SAVE
        NOSAVE
```

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 not specify UPDATE on the READ command, then the NOSAVE option is forced.

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 the READ UPDATE function

```
REREAD
```

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
```

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.

Examples using BLKUPD (BLKUPD)

Note: This example illustrates the functions of Block Update. It is not meant to be replicated for purposes of repairing your RACF database.

BLKUPD:

- Issue the BLKUPD command on the RACF database that you want to work with.

```
BLKUPD 'SYS1.RACF'
BLKUPD:
```

- 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  AARDVARK                                  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
```

- 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.
- 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.
  ```
- List the modified area of the new block (8 bytes beginning at offset X'9A').

  ```
  LIST NEW RANGE(X'9A',8)
  RBA=00000000E000
    009A D4F2F7F0 01010000
    *M270....                       *
  ```
- Delete the first entry in the index block.

  ```
  BLKUPD:
  DISPLAY
   OFFSET COMP.       ENTRY NAME                                  RBA
          COUNT
    00E    000    AARDVARK                                  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
```

- Insert a new first index entry.

```
INSERT ENTRY(ANT) 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    ANT                                            00000000DC00
NEXT
```

displays the next index entry.

```
   00E    000    HILDE                                          00000000D000
                                          SEGMENT NAME: TSO    00000000D900
                                                        DFP    00000000DA00
```

- 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
```

- 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 executes under the READ command.

```
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  ANT                                   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
```

- Save the changes made and have them written 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 data base.
- Locate the block that contains the generic name of SALES.*.

  ```
  LOCATE ENTRY(SALES.*) CLASS(DATASET) GENERIC
  ```

  prints 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  ANT                                   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
```

- End the BLKUPD session.

```
BLKUPD:
END
READY
```

ends the BLKUPD utility.

Chapter 6. Parameter Lists

This section describes parameter lists passed to RACF when a RACF macro is invoked. Installation exits also use parameter lists. These are described in *z/VM: RACF Security Server System Programmer's Guide*.

The parameter lists used by RACF in processing RACF commands and the RACF report writer are standard TSO/E parameter lists. See *TSO/E Customization* for information on these parameter lists.

The parameter lists used by RACF in processing RACF utilities and DSMON are standard z/OS parameter lists.

Table 18 is a quick reference that identifies the system macro-request types that are replacements for the independent RACF system macros. For more information on both types, see *z/VM: Security Server RACROUTE Macro Reference*.

Table 18. Cross-Reference for RACROUTE REQUEST=type, the Independent RACF System Macros, and the Parameter List Name

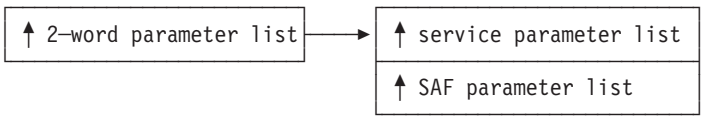
RACROUTE REQUEST=*type*	**Independent RACF System Macro**	**Parameter List**
REQUEST=AUDIT	RACAUDIT	AUDIT
REQUEST=AUTH	RACHECK	AUTH
REQUEST=DEFINE	RACDEF	DEFINE
REQUEST=DIRAUTH	None	DIRAUTH
REQUEST=EXTRACT	RACXTRT	EXTRACT
REQUEST=FASTAUTH	FRACHECK	FASTAUTH
REQUEST=LIST	RACLIST	LIST
REQUEST=STAT	RACSTAT	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 in this section:

"RACROUTE Parameter List"
"AUDIT service parameter list" on page 91
"AUTH service parameter list" on page 92
"DEFINE service parameter list" on page 96
"DIRAUTH Service Parameter List" on page 102
"EXTRACT service parameter list" on page 103
"FASTAUTH service parameter list" on page 105
"LIST service parameter list" on page 106
"STAT service parameter list" on page 107
"TOKENBLD service parameter list" on page 108
"TOKENMAP/TOKENXTR service parameter list" on page 112
"VERIFY/VERIFYX service parameter list" on page 113
"ICHEINTY parameter list" on page 117

"Checking the Parameter List Passed to RACF Macros on z/VM" on page 41 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. It is mapped by macro ICHSAFP. For another description of this parameter list that includes the field names, see *z/VM: Security Server RACROUTE Macro Reference*.

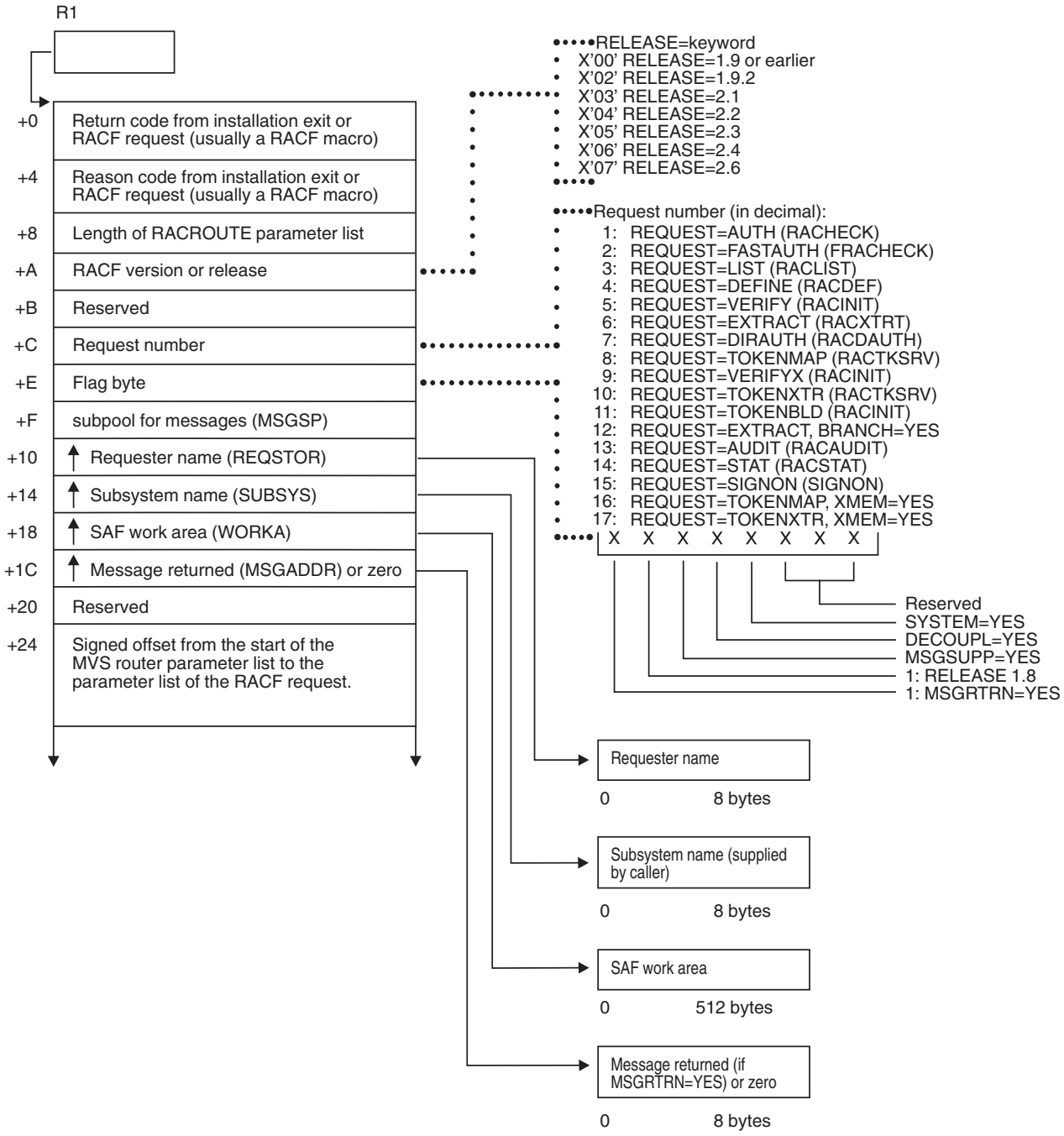


Figure 12. RACROUTE Parameter List (Part 1 of 2)

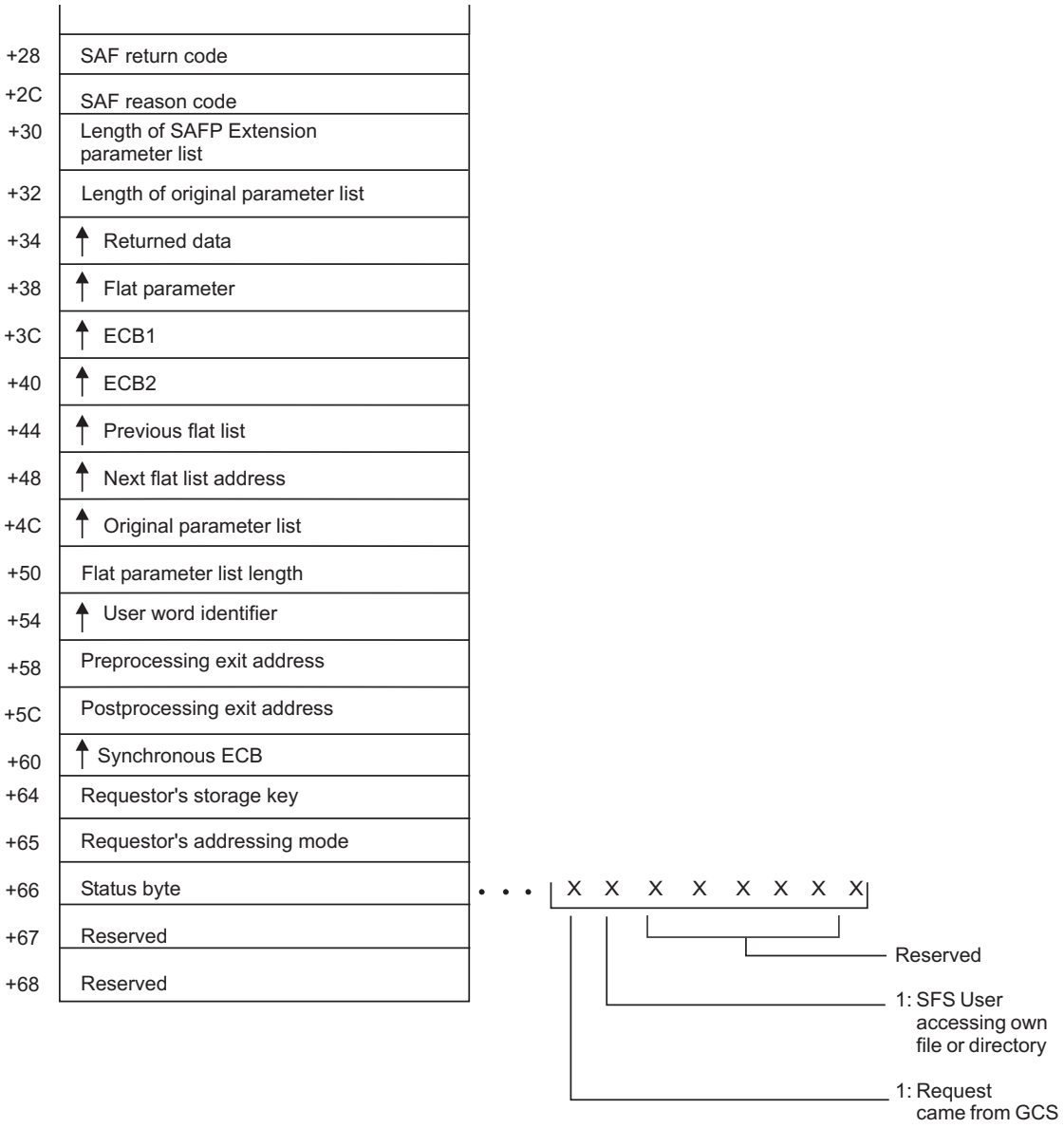


Figure 12. RACROUTE Parameter List (Part 2 of 2)

AUDIT service parameter list

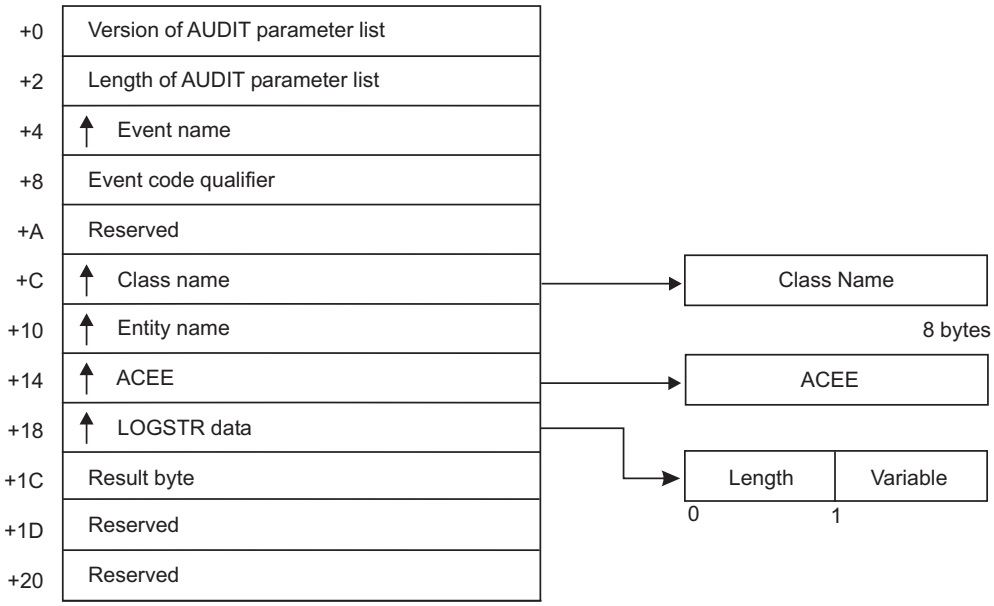
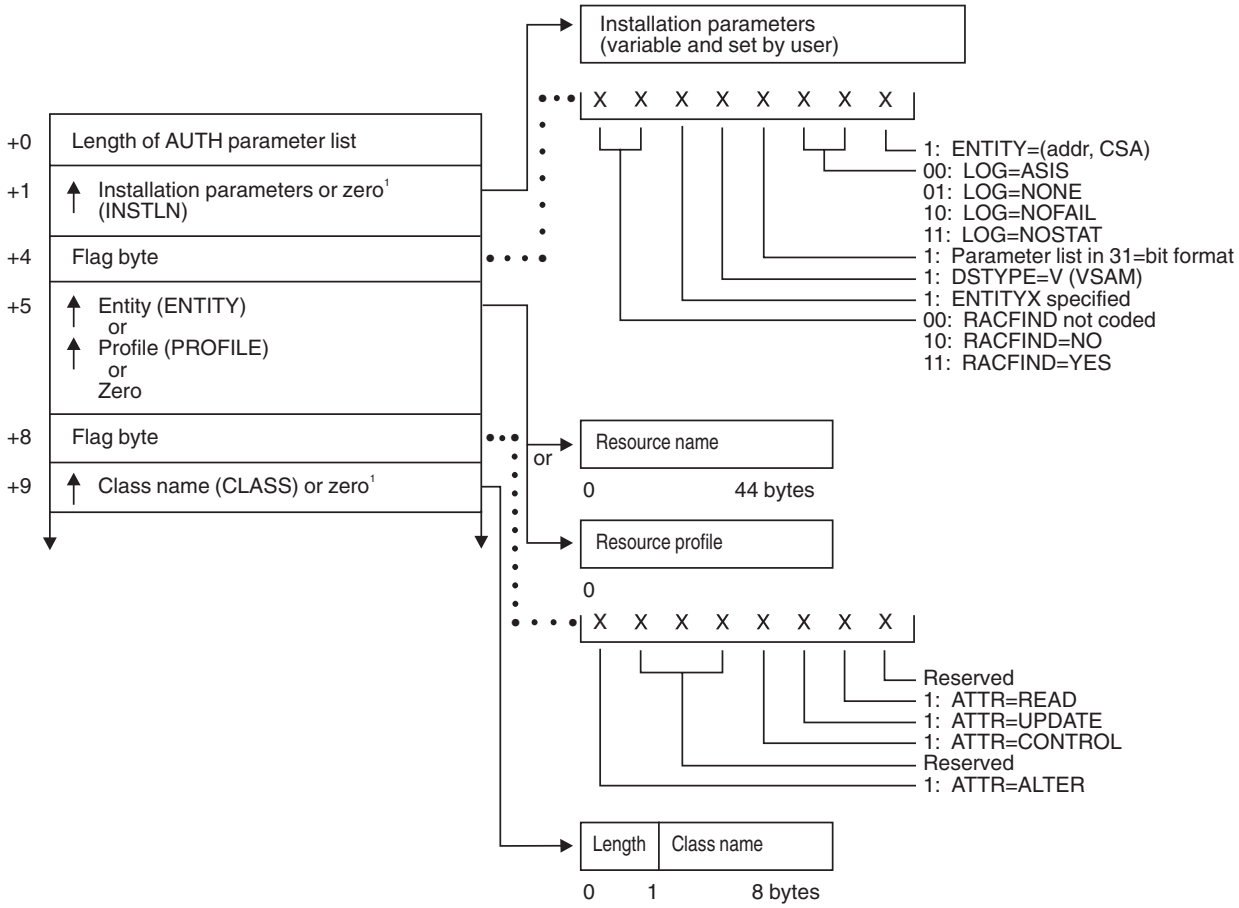


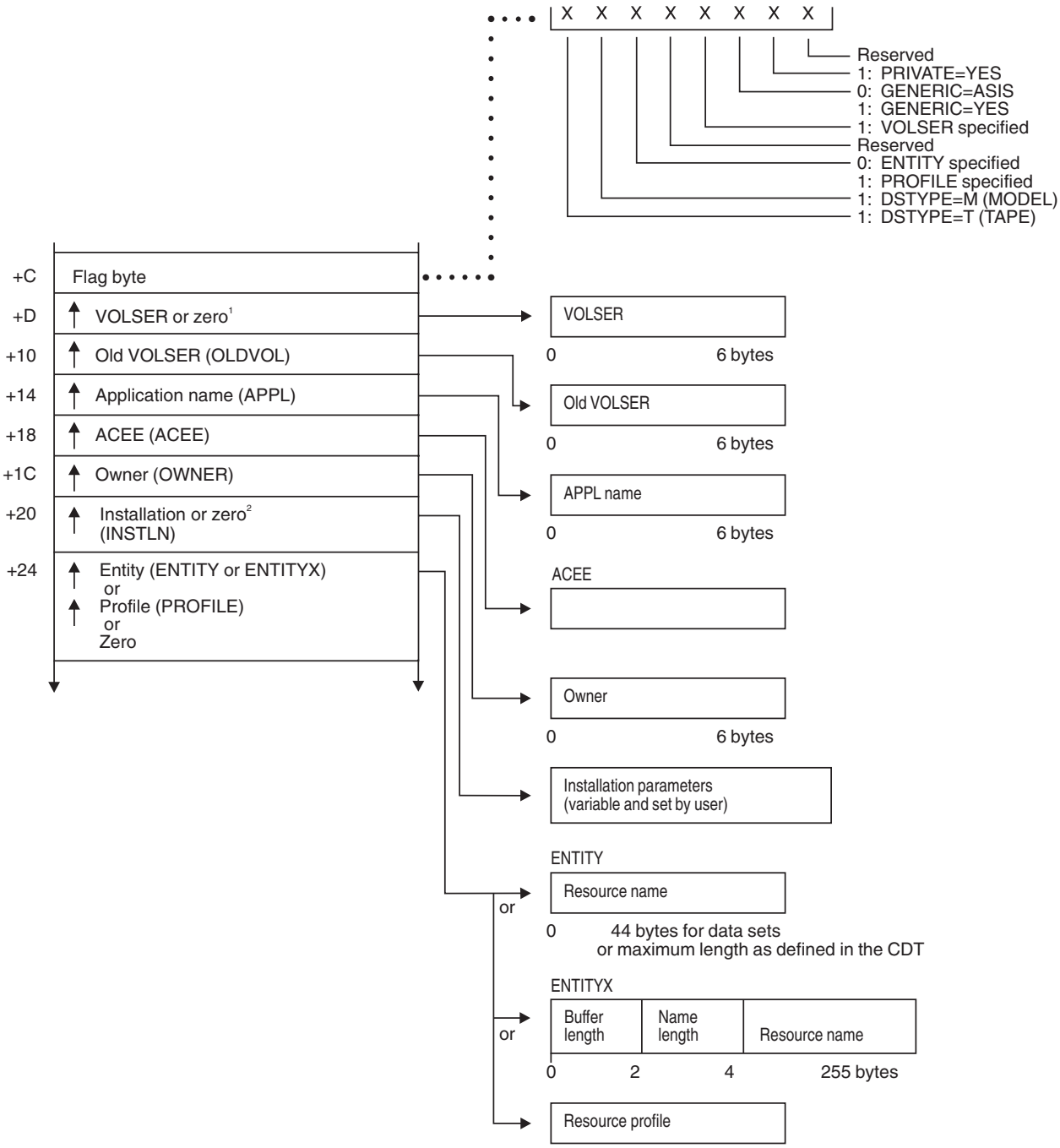
Figure 13. AUDIT Service Parameter List

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).

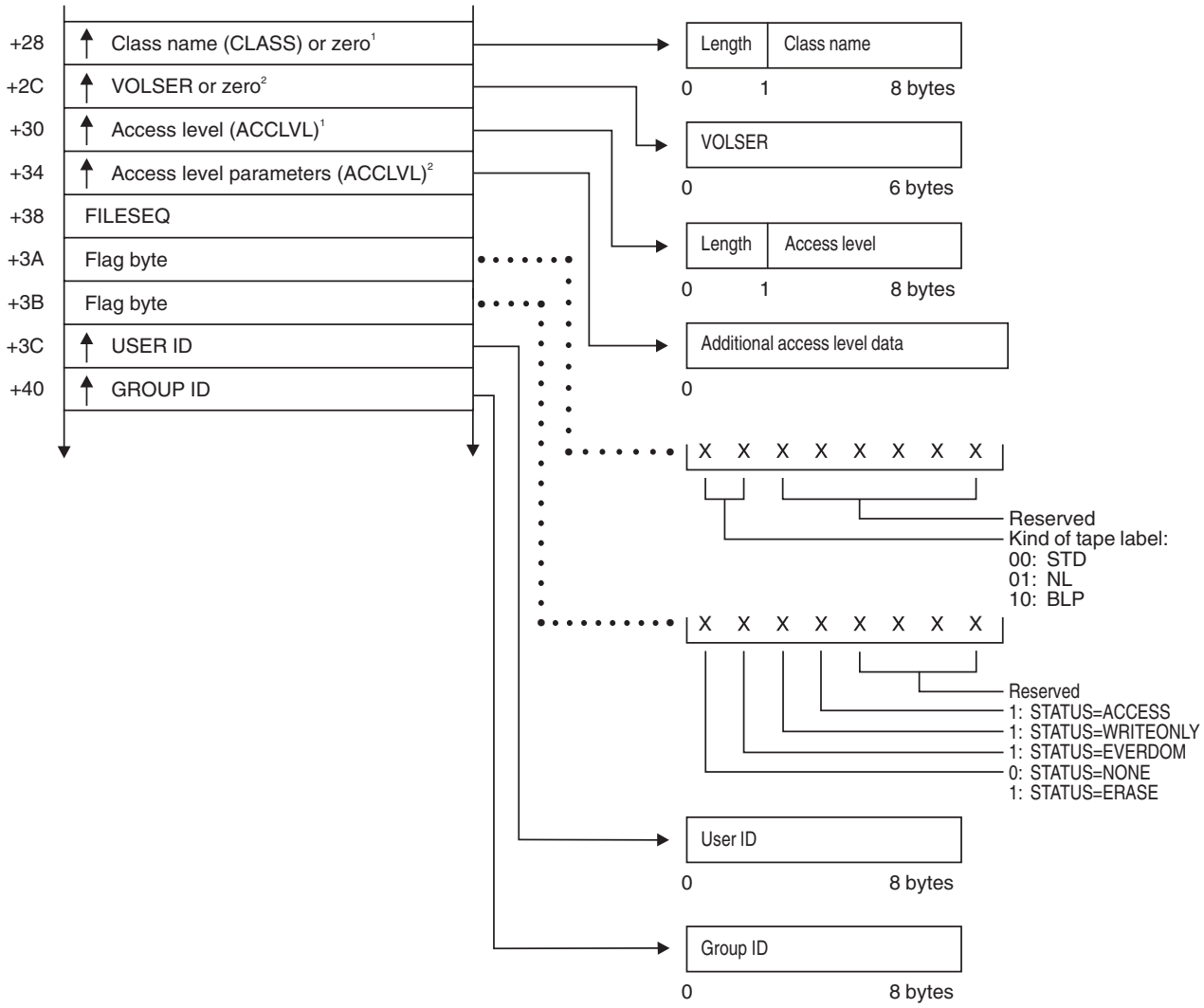
Figure 14. AUTH Service Parameter List (Part 1 of 4)



[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 14. AUTH Service Parameter List (Part 2 of 4)

AUTH parameters



[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 14. AUTH Service Parameter List (Part 3 of 4)

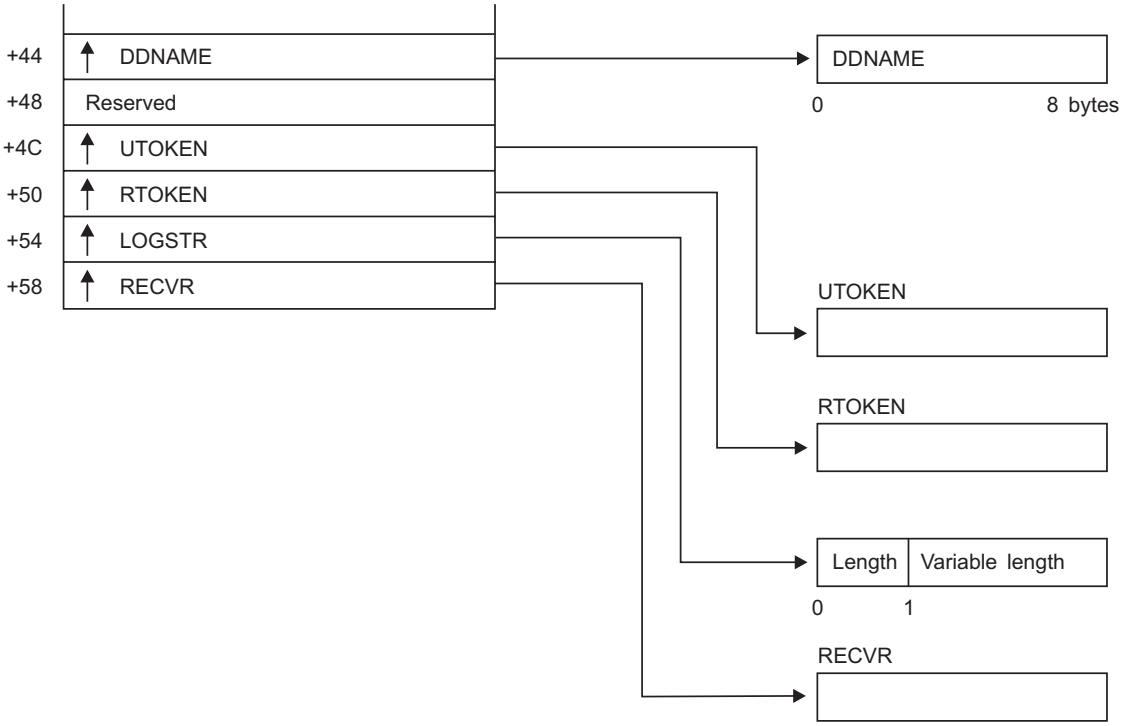
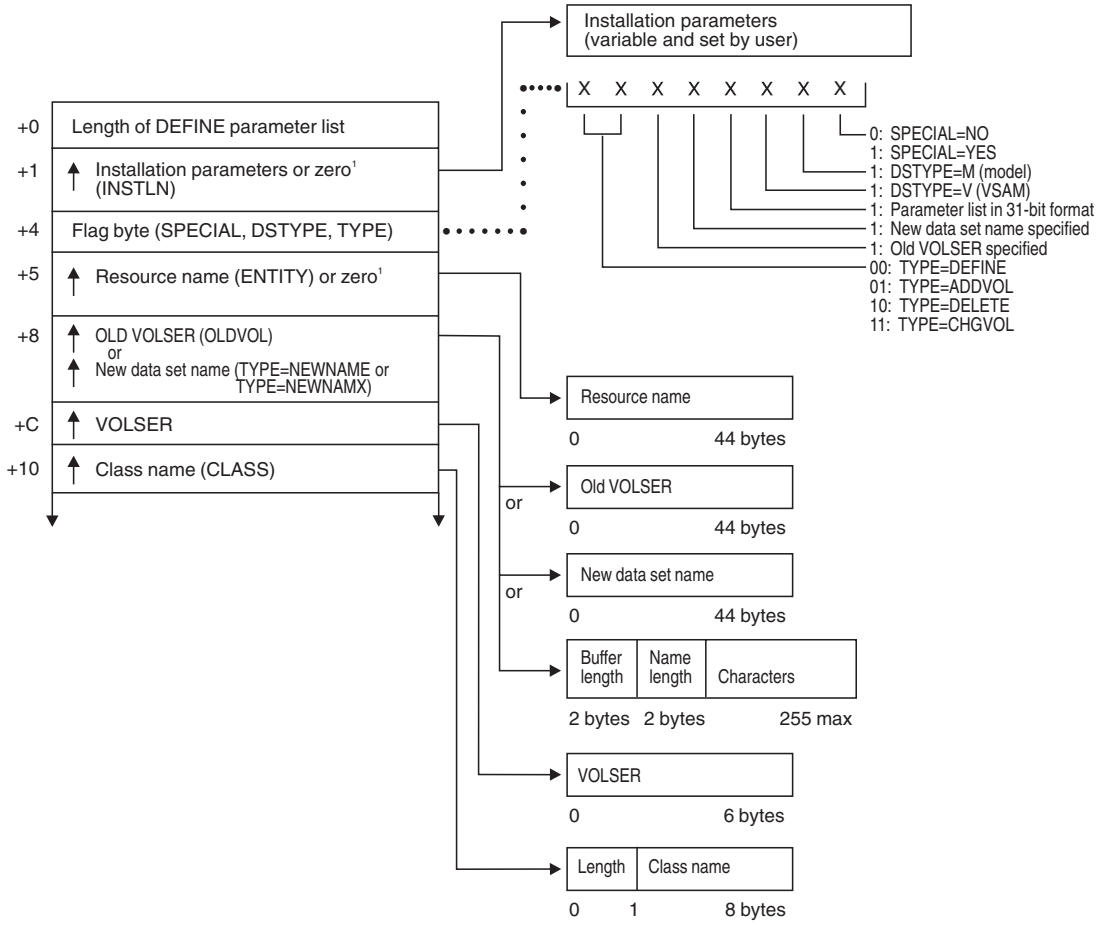


Figure 14. AUTH Service Parameter List (Part 4 of 4)

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 15. DEFINE Service Parameter List (Part 1 of 6)

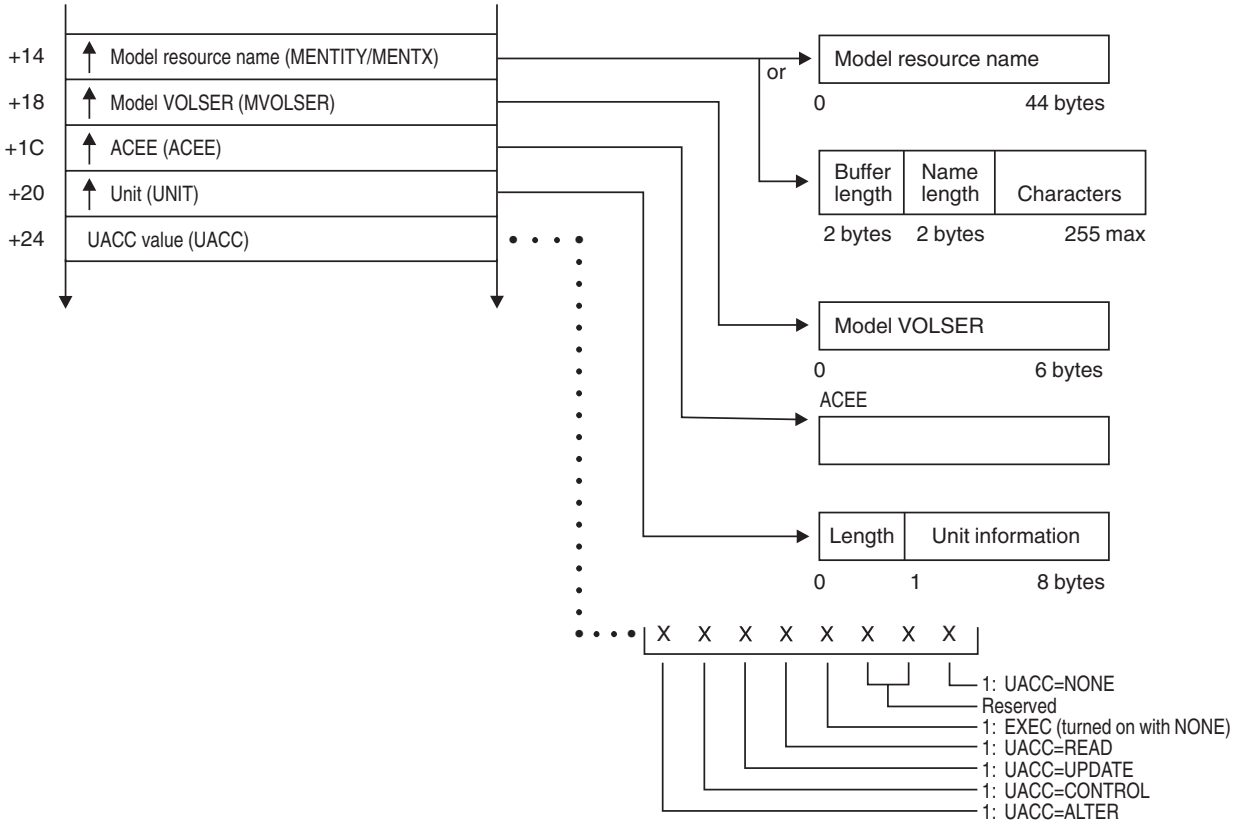


Figure 15. DEFINE Service Parameter List (Part 2 of 6)

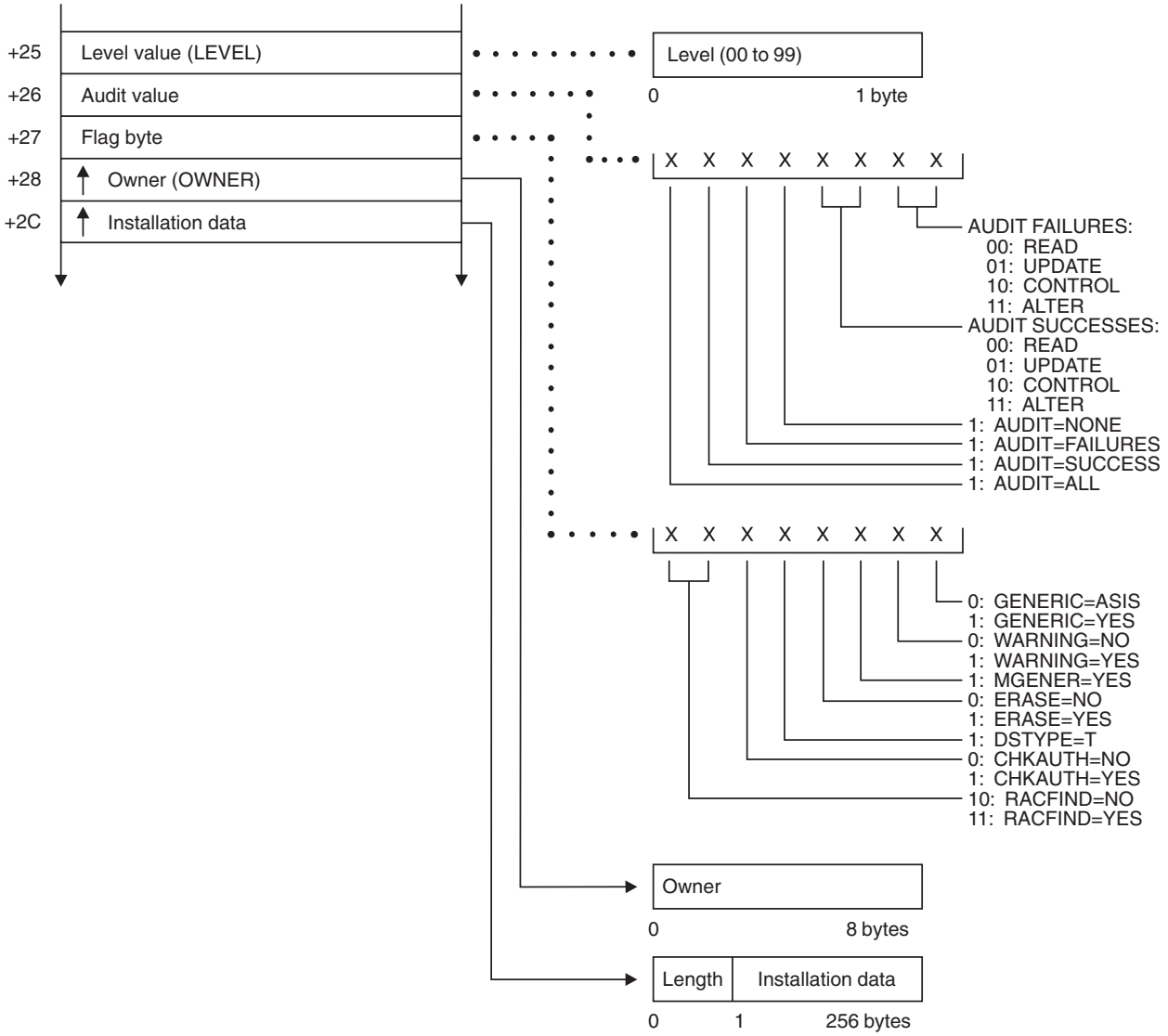
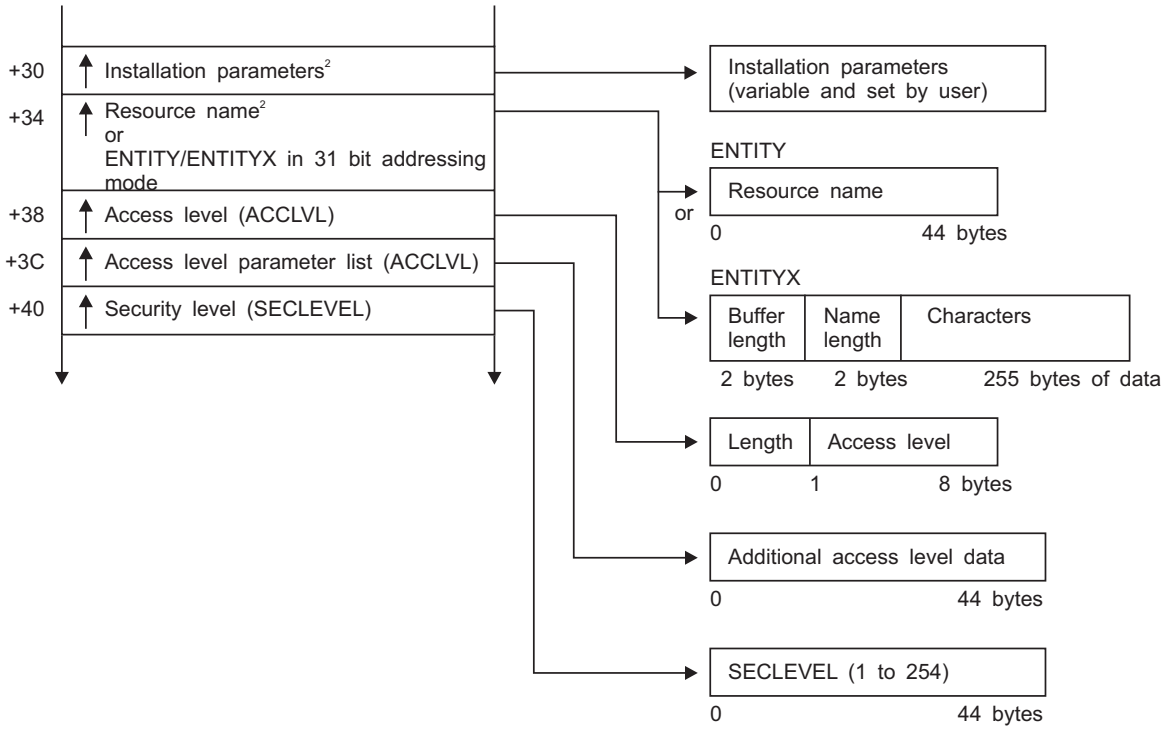


Figure 15. DEFINE Service Parameter List (Part 3 of 6)



[2] This field is zero if bit 4 of the flag byte at offset 4 is zero (input was RACDEF macro).

Figure 15. DEFINE Service Parameter List (Part 4 of 6)

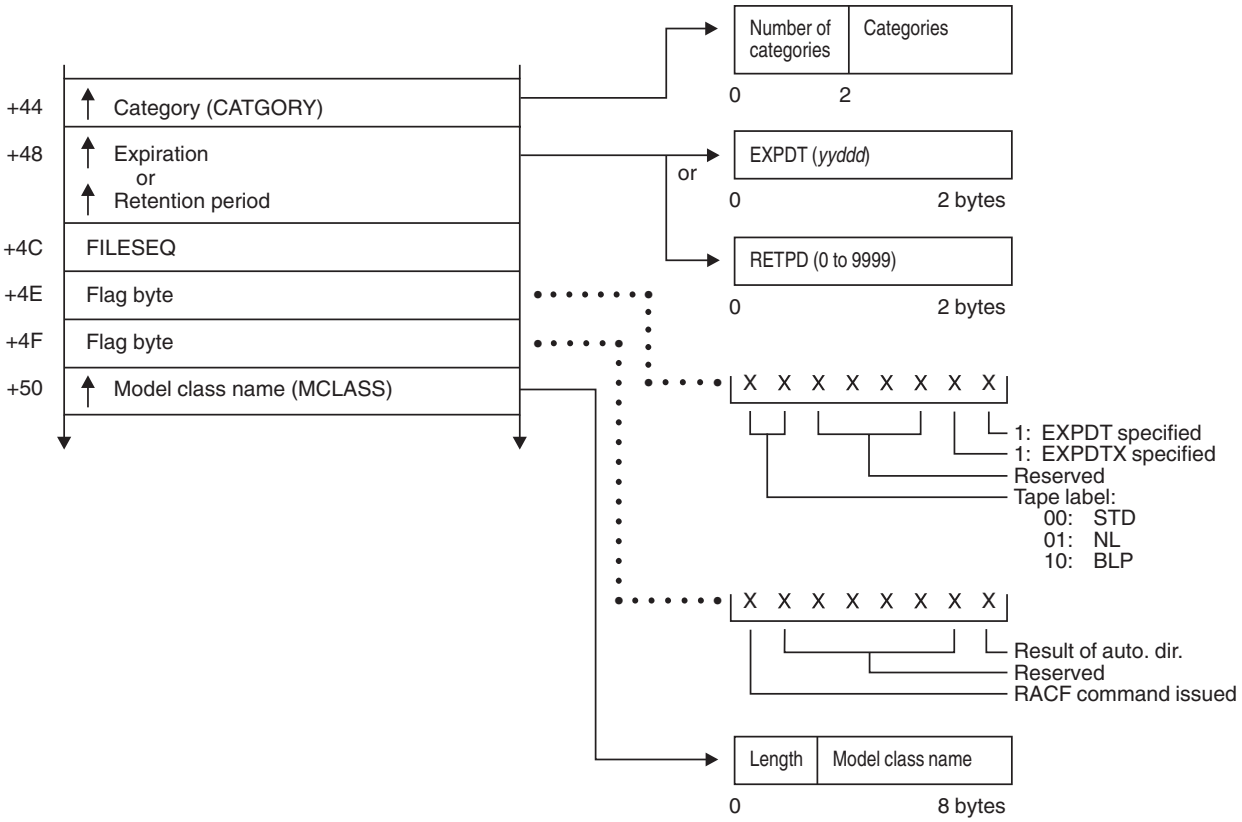


Figure 15. DEFINE Service Parameter List (Part 5 of 6)

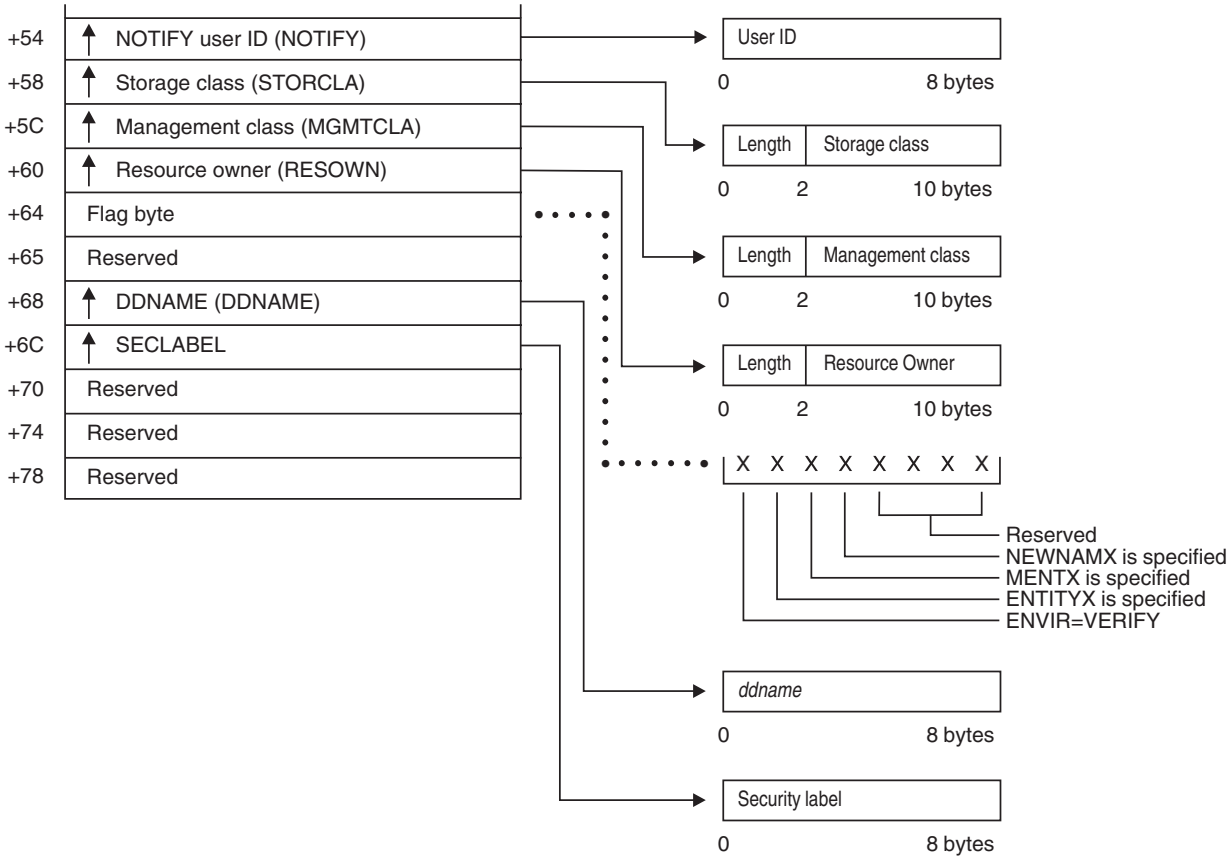


Figure 15. DEFINE Service Parameter List (Part 6 of 6)

DIRAUTH Service Parameter List

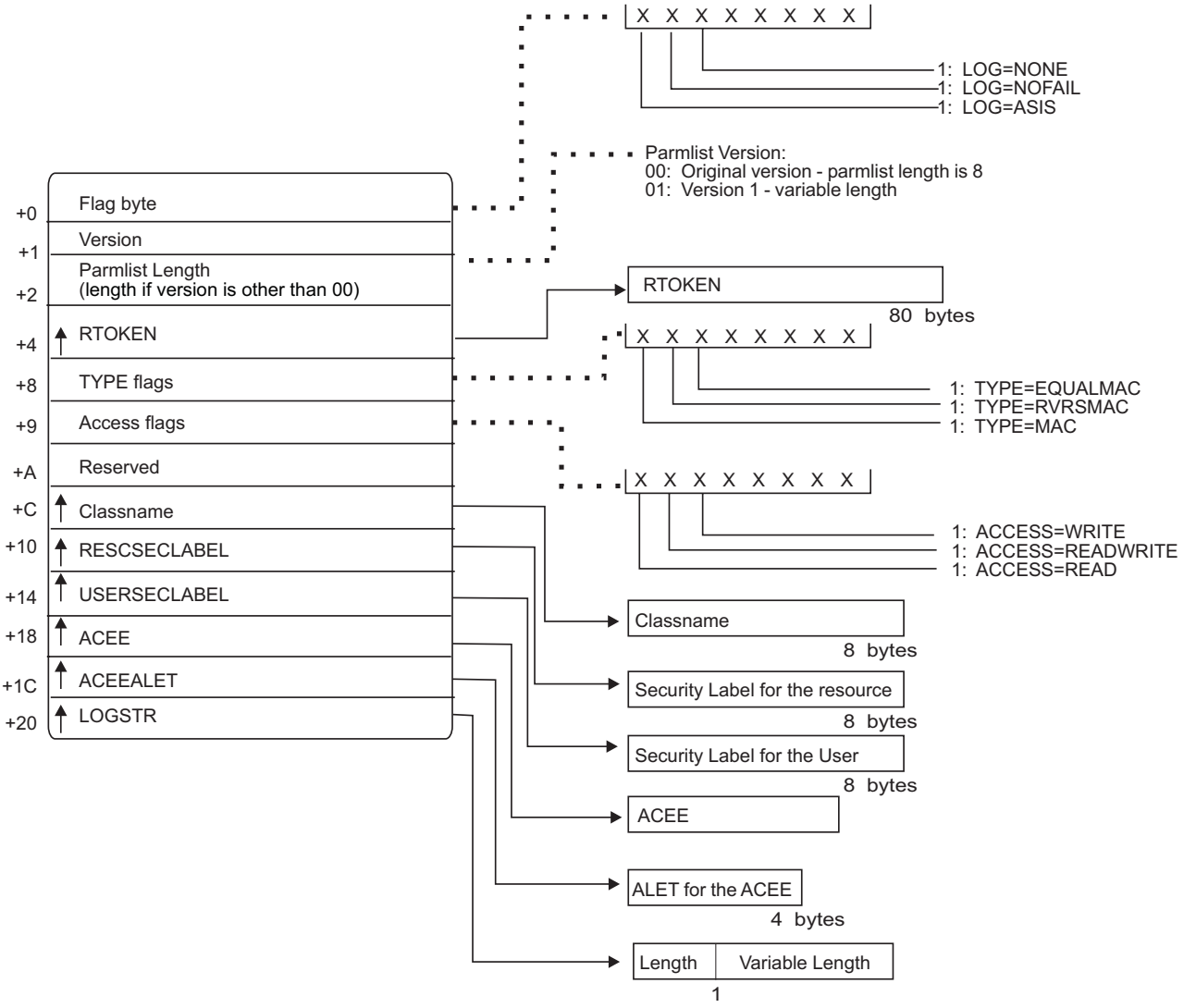


Figure 16. DIRAUTH service parameter list

EXTRACT service parameter list

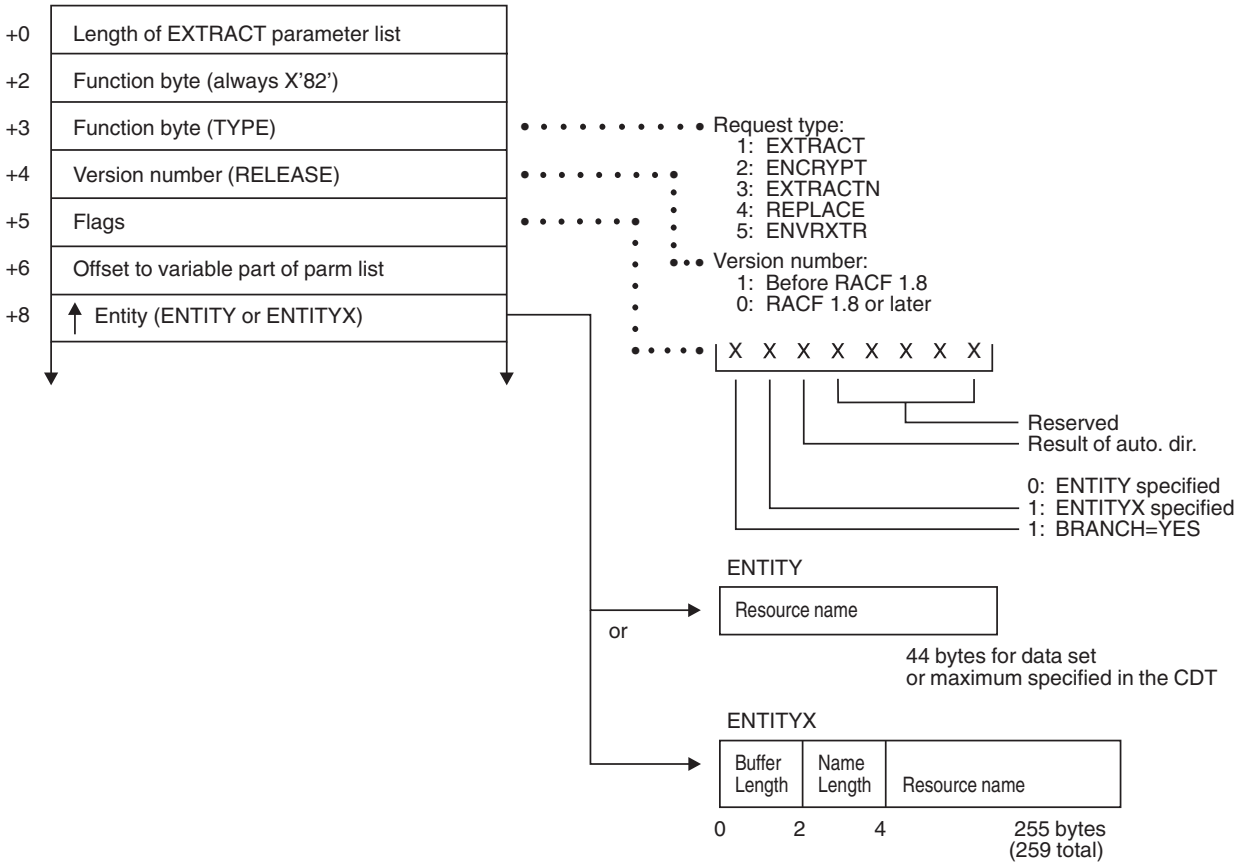


Figure 17. EXTRACT Service Parameter List (Part 1 of 3)

(The following is used when TYPE = ENVRXTR.)

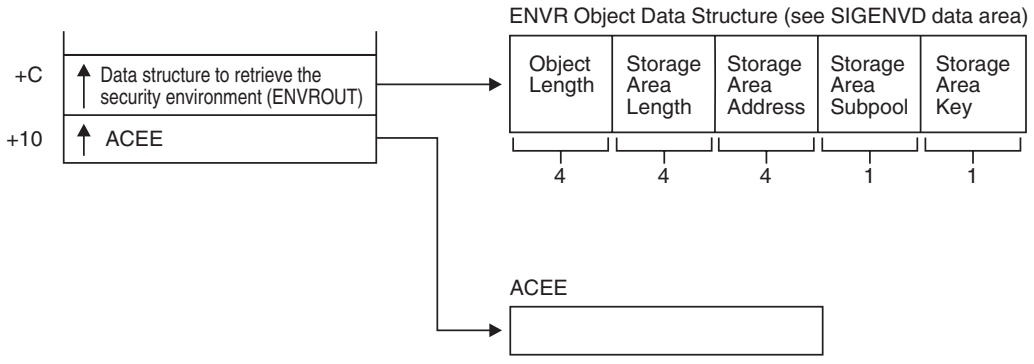
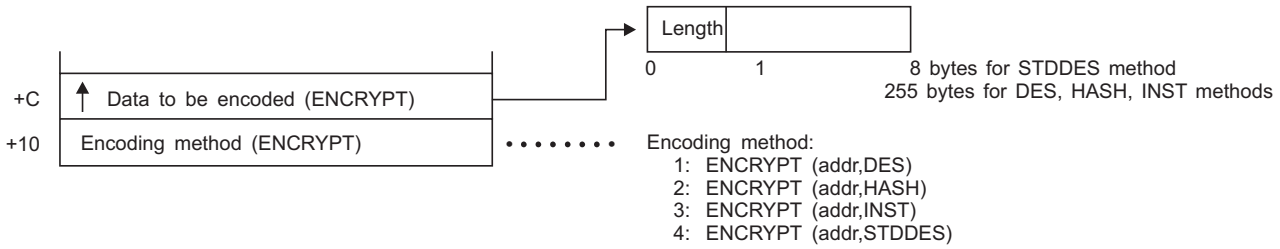


Figure 17. EXTRACT Service Parameter List (Part 2 of 3)

(The following is used when TYPE=ENCRYPT.)



(The following is used when TYPE is not ENCRYPT or ENVRXTR.)

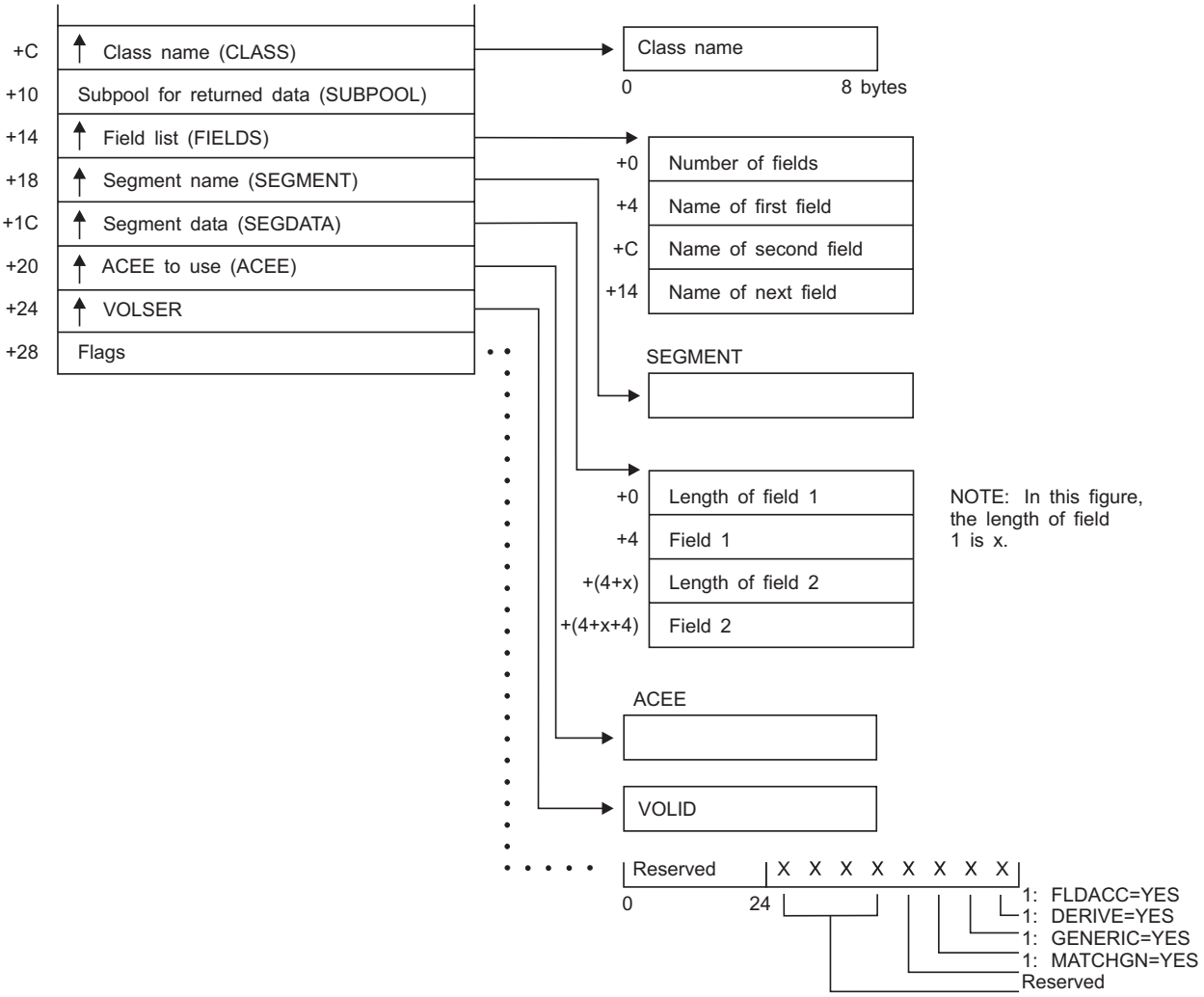


Figure 17. EXTRACT Service Parameter List (Part 3 of 3)

FASTAUTH service parameter list

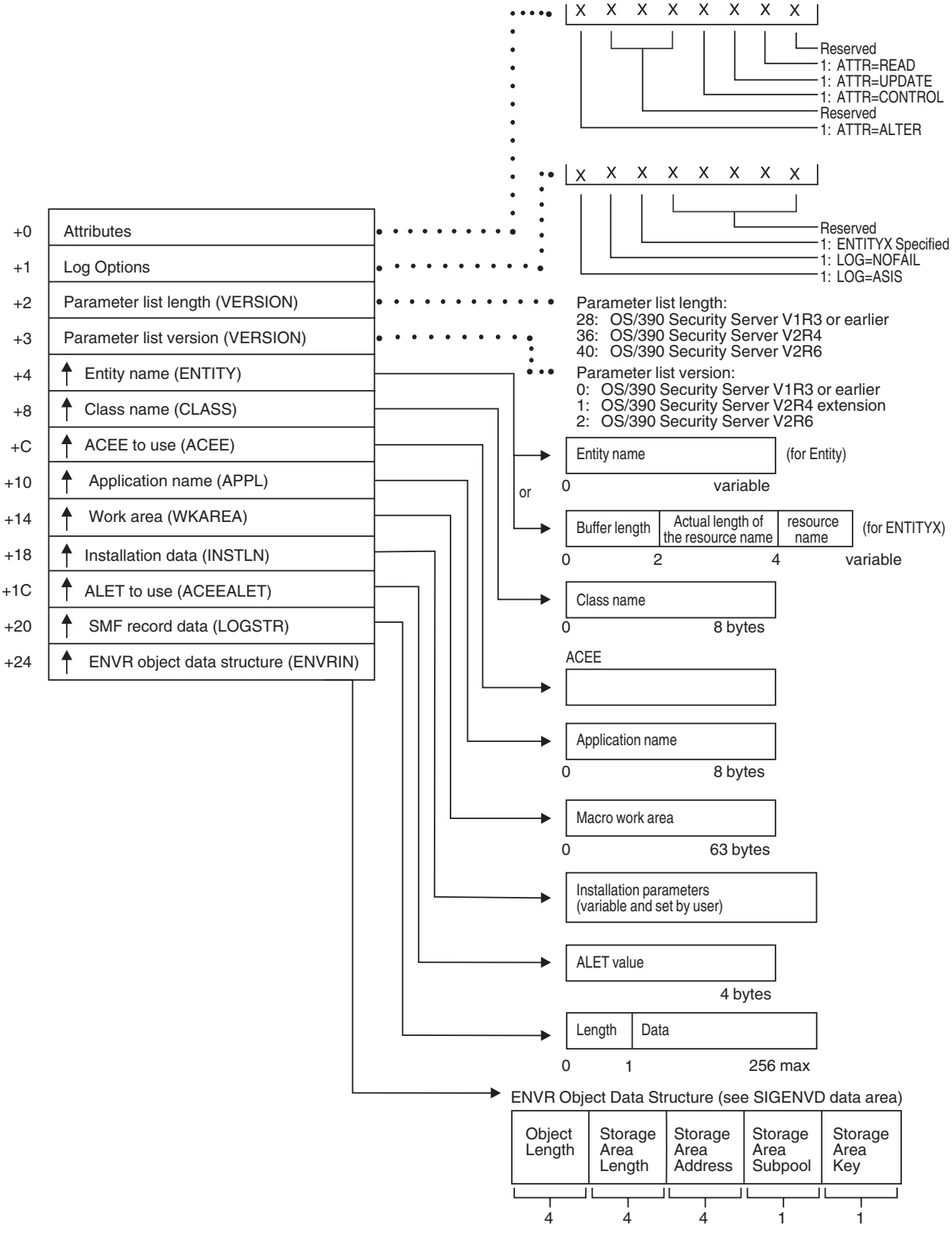


Figure 18. FASTAUTH Service Parameter List

LIST service parameter list

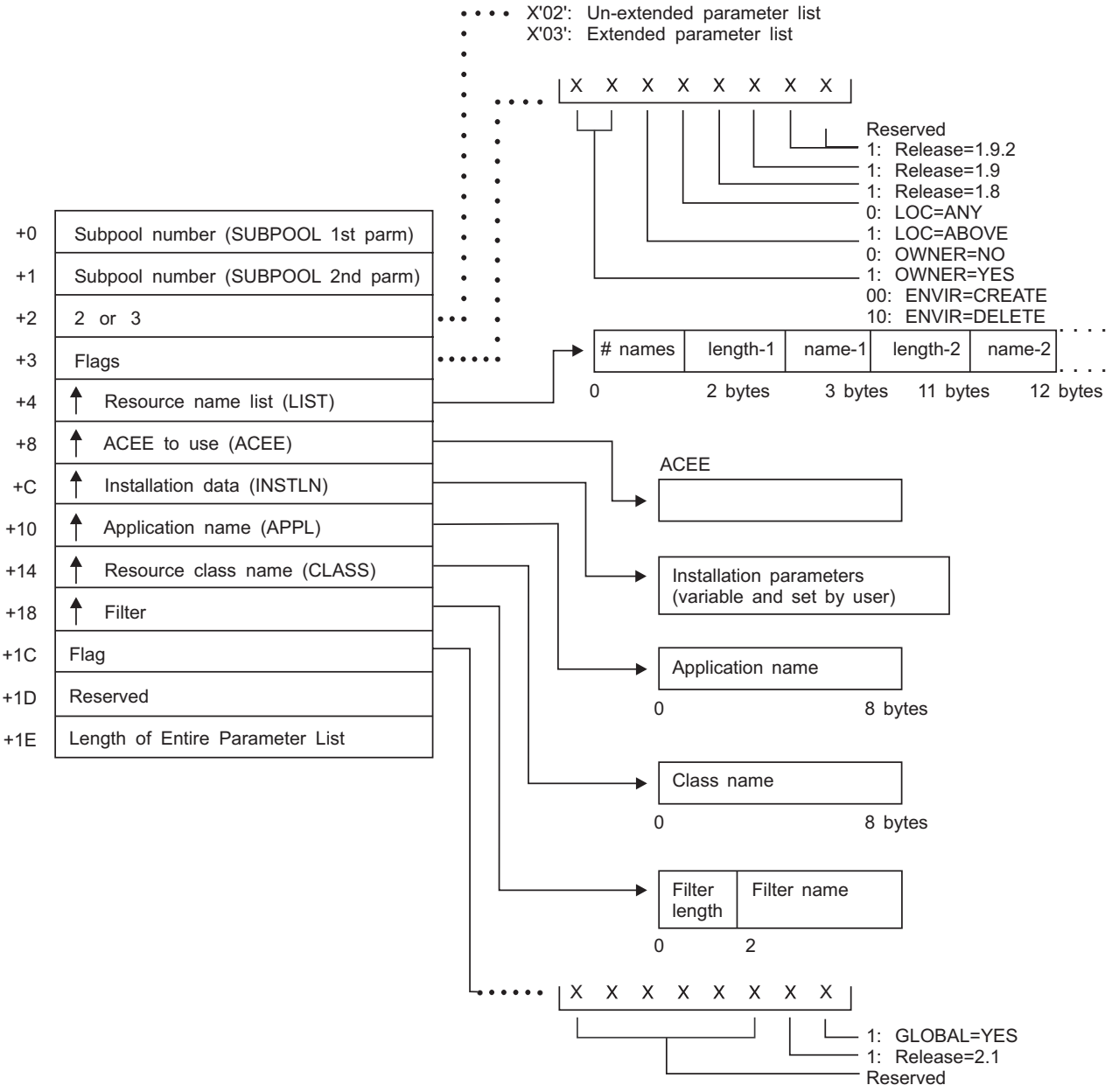


Figure 19. LIST Service Parameter List

STAT service parameter list

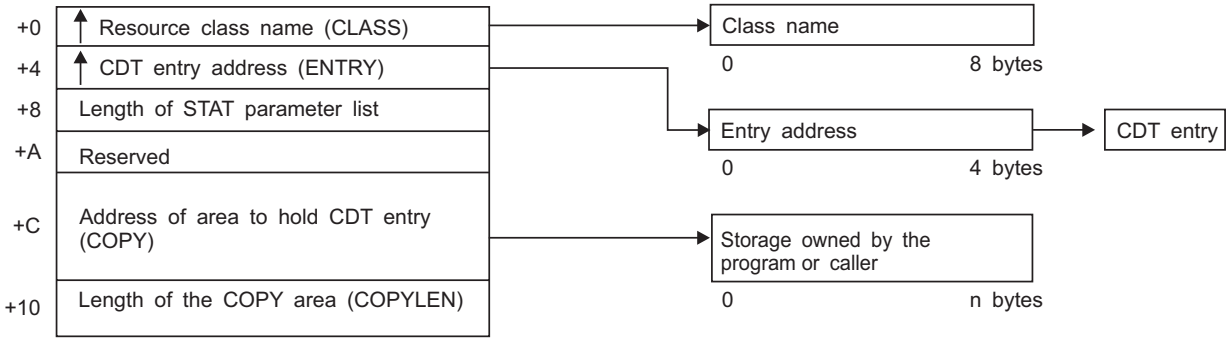


Figure 20. STAT Service Parameter List

TOKENBLD service parameter list

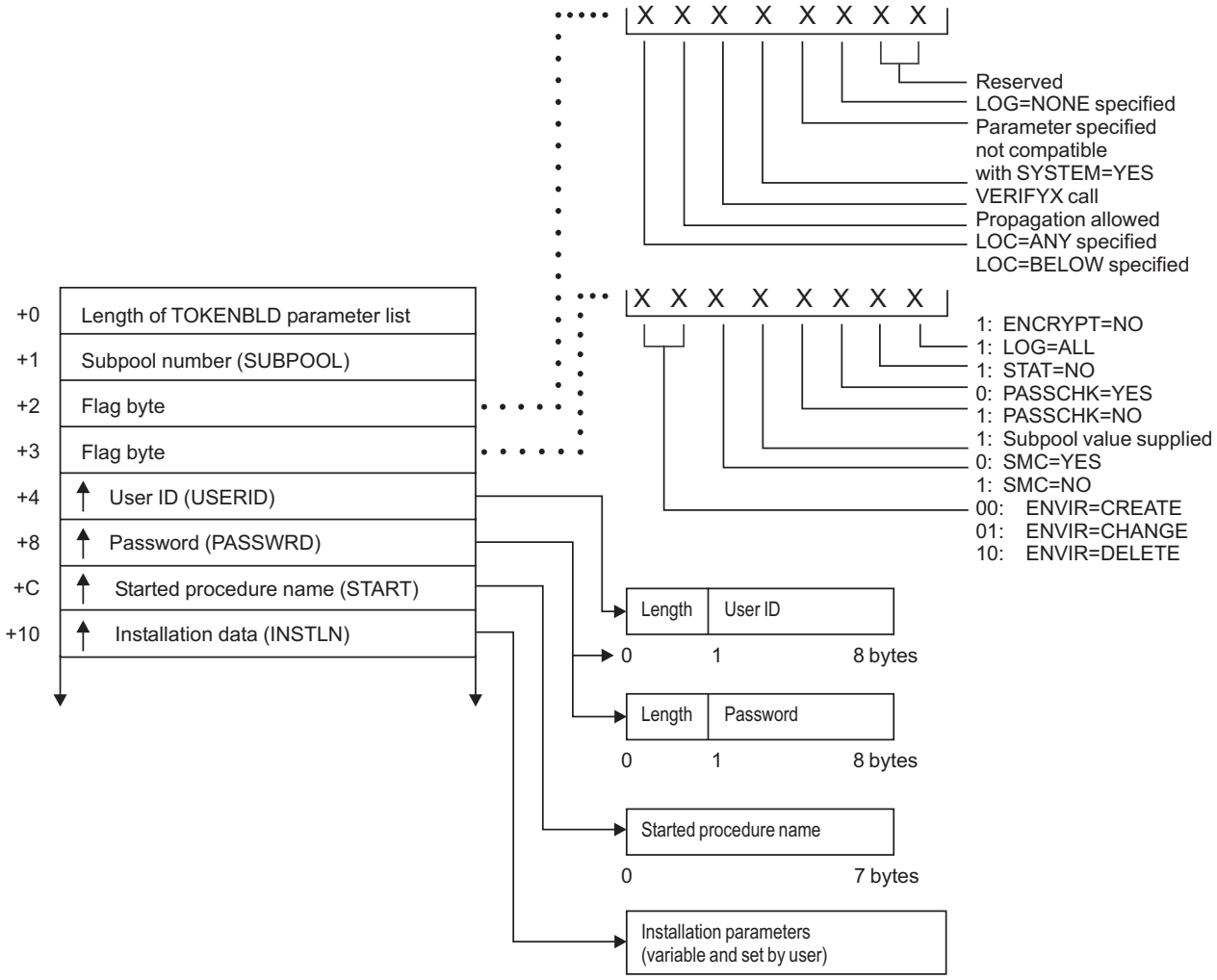


Figure 21. TOKENBLD Service Parameter List (Part 1 of 4)

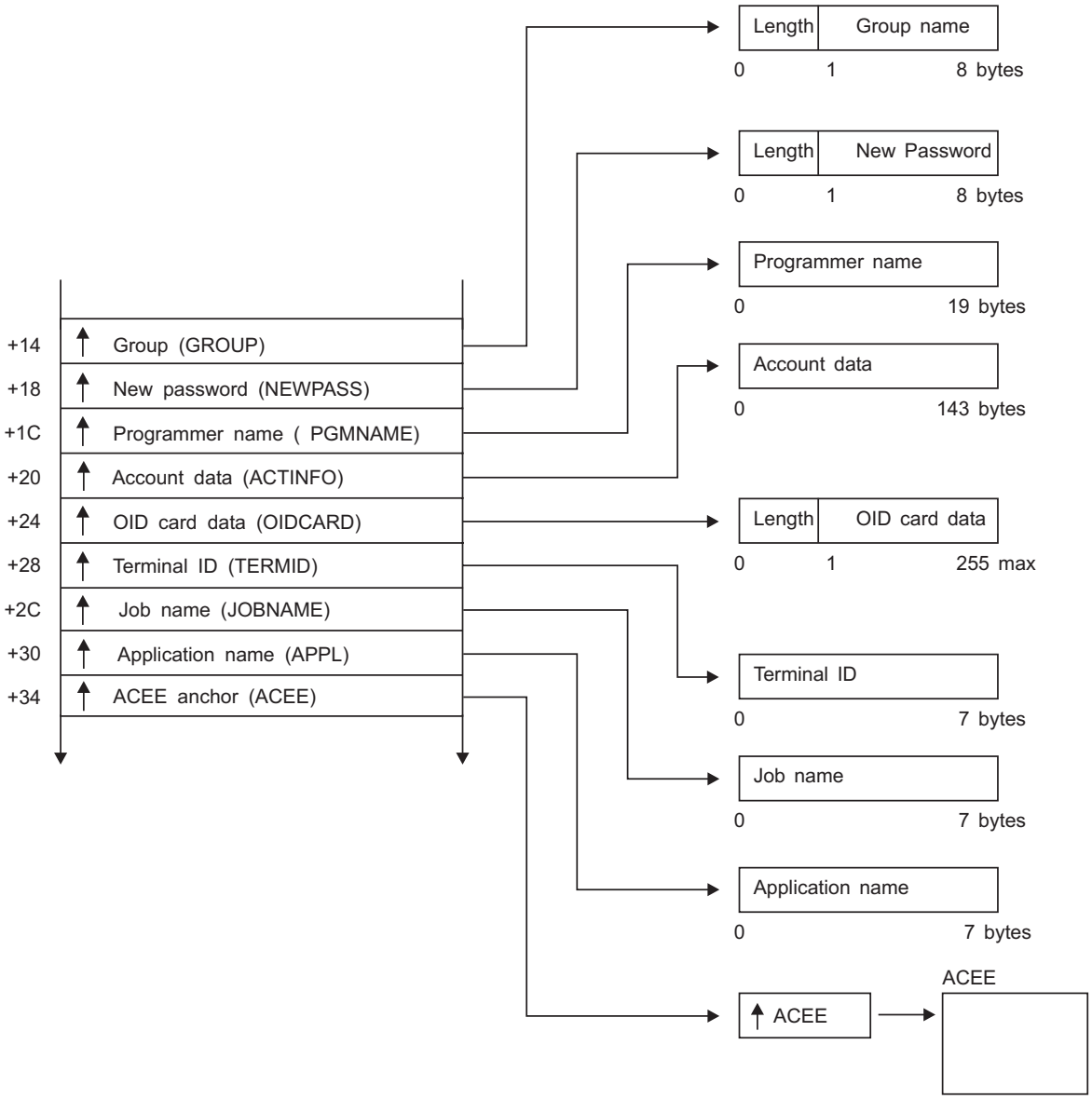


Figure 21. TOKENBLD Service Parameter List (Part 2 of 4)

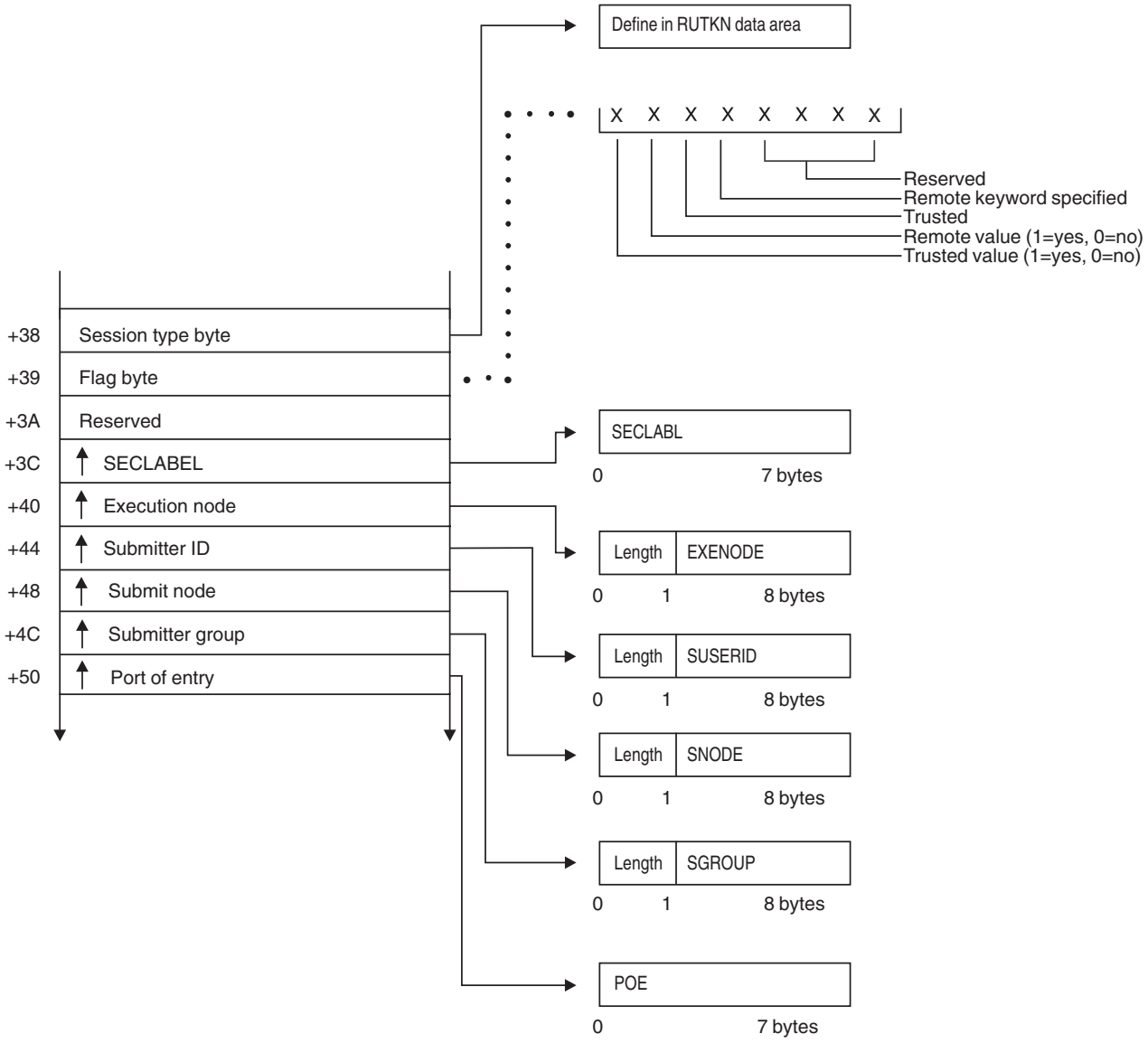


Figure 21. TOKENBLD Service Parameter List (Part 3 of 4)

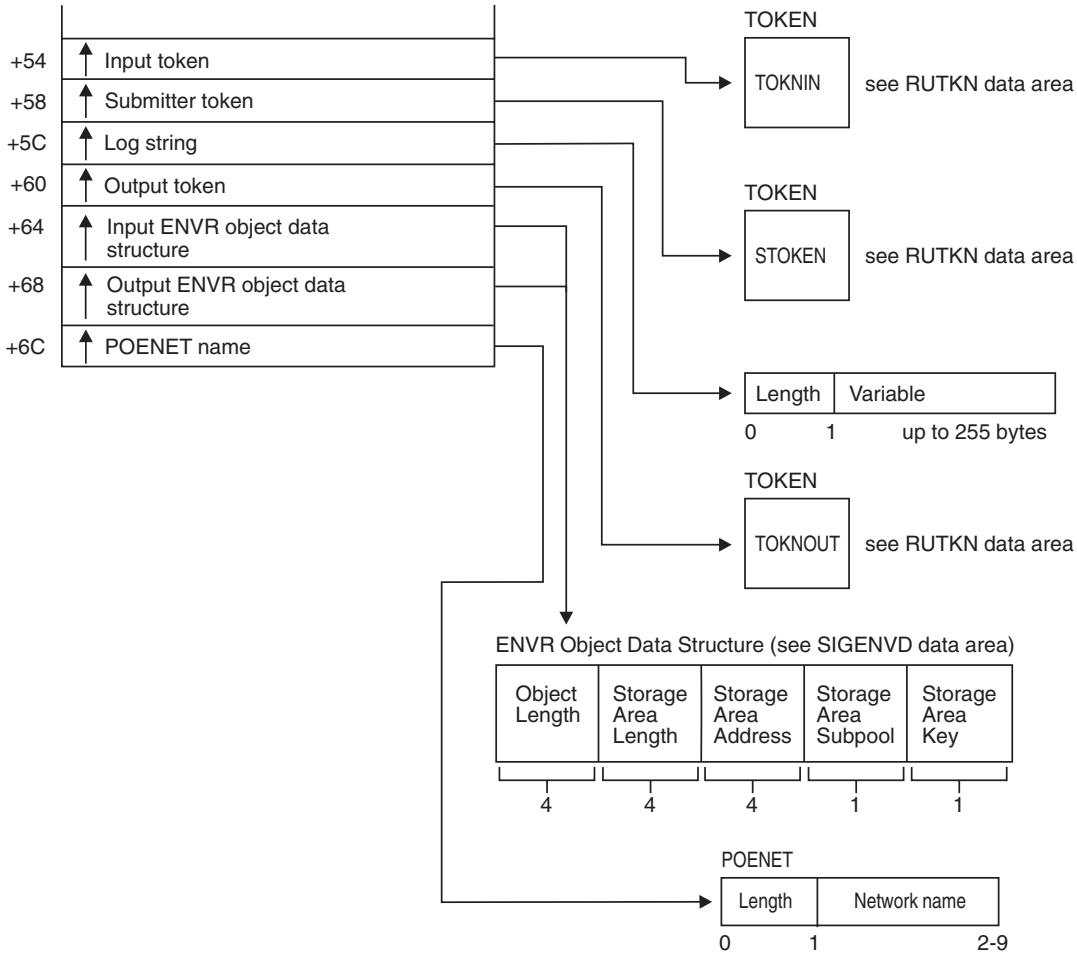


Figure 21. TOKENBLD Service Parameter List (Part 4 of 4)

TOKENMAP/TOKENXTR service parameter list

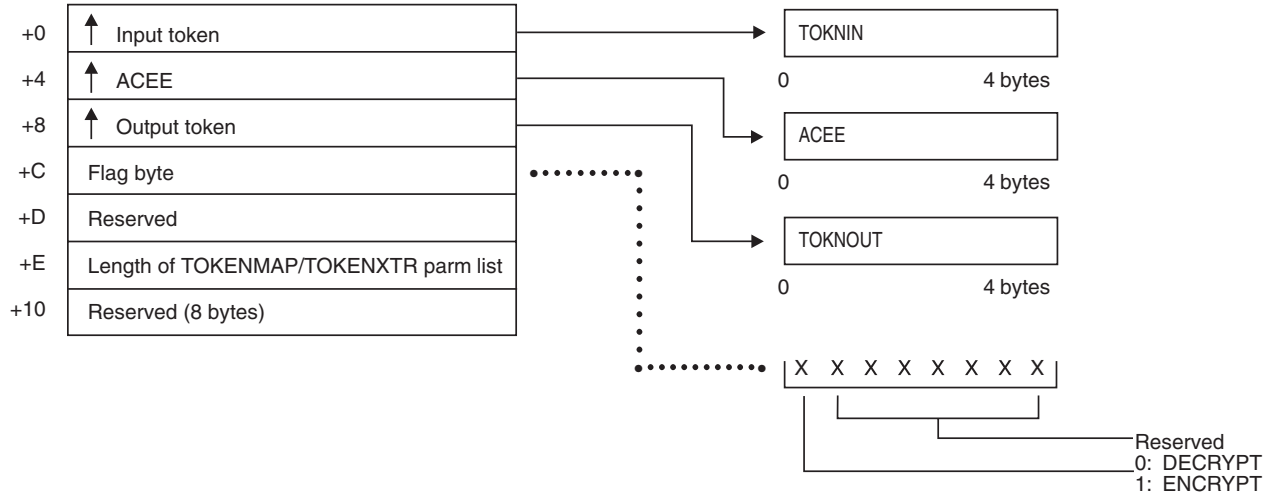


Figure 22. TOKENMAP/TOKENXTR Service Parameter List

VERIFY/VERIFYX service parameter list

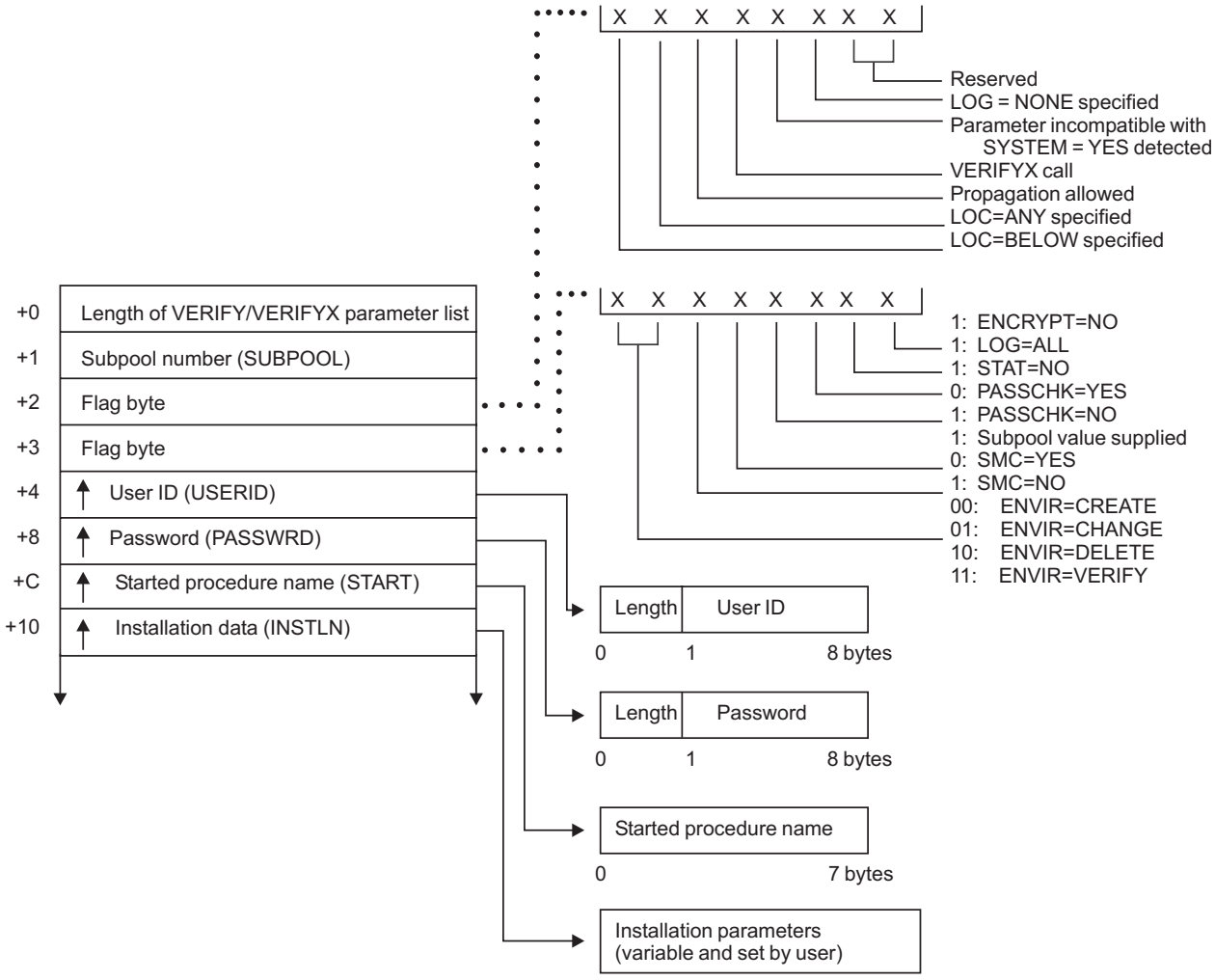


Figure 23. VERIFY/VERIFYX Service Parameter List (Part 1 of 4)

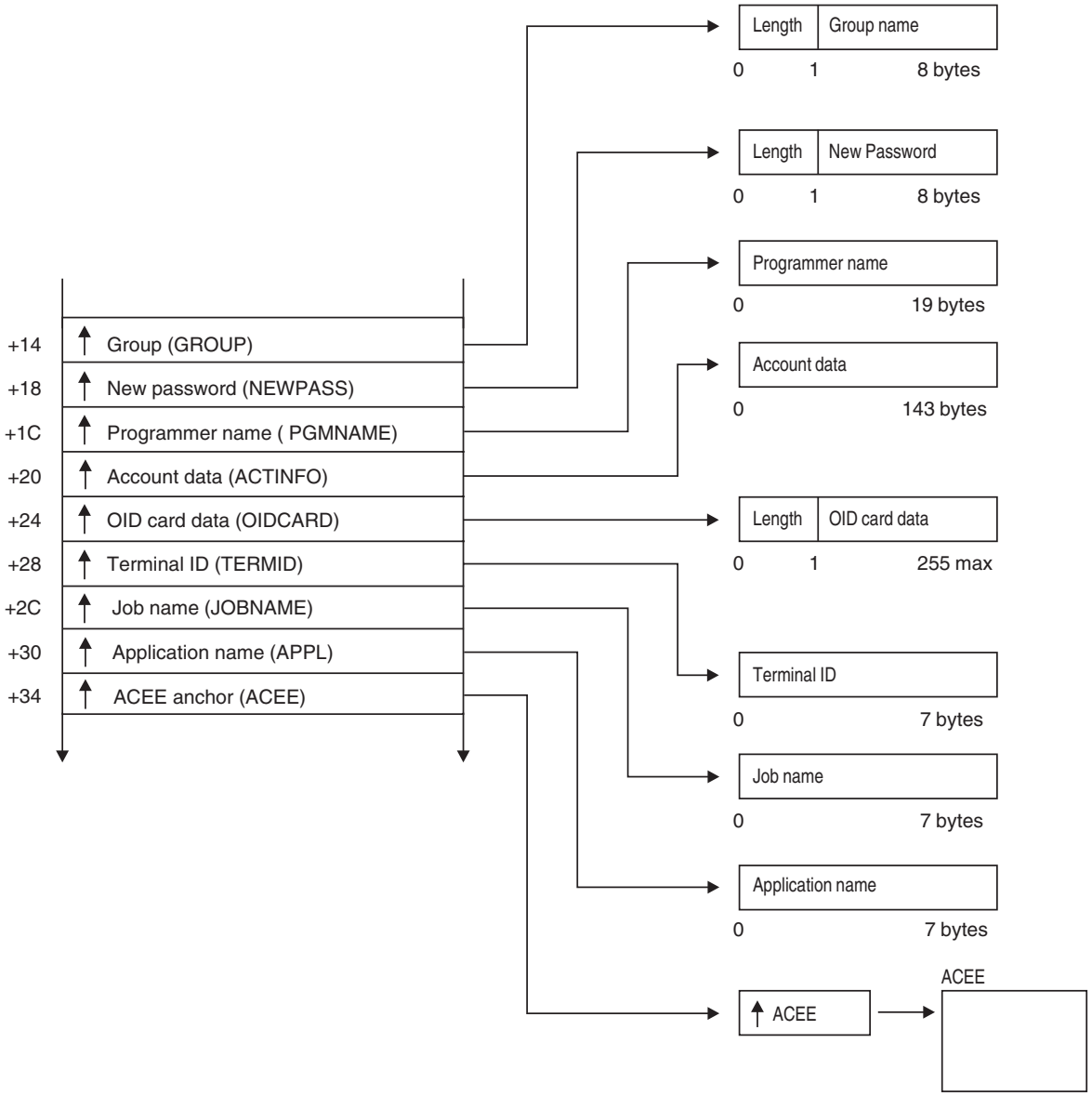


Figure 23. VERIFY/VERIFYX Service Parameter List (Part 2 of 4)

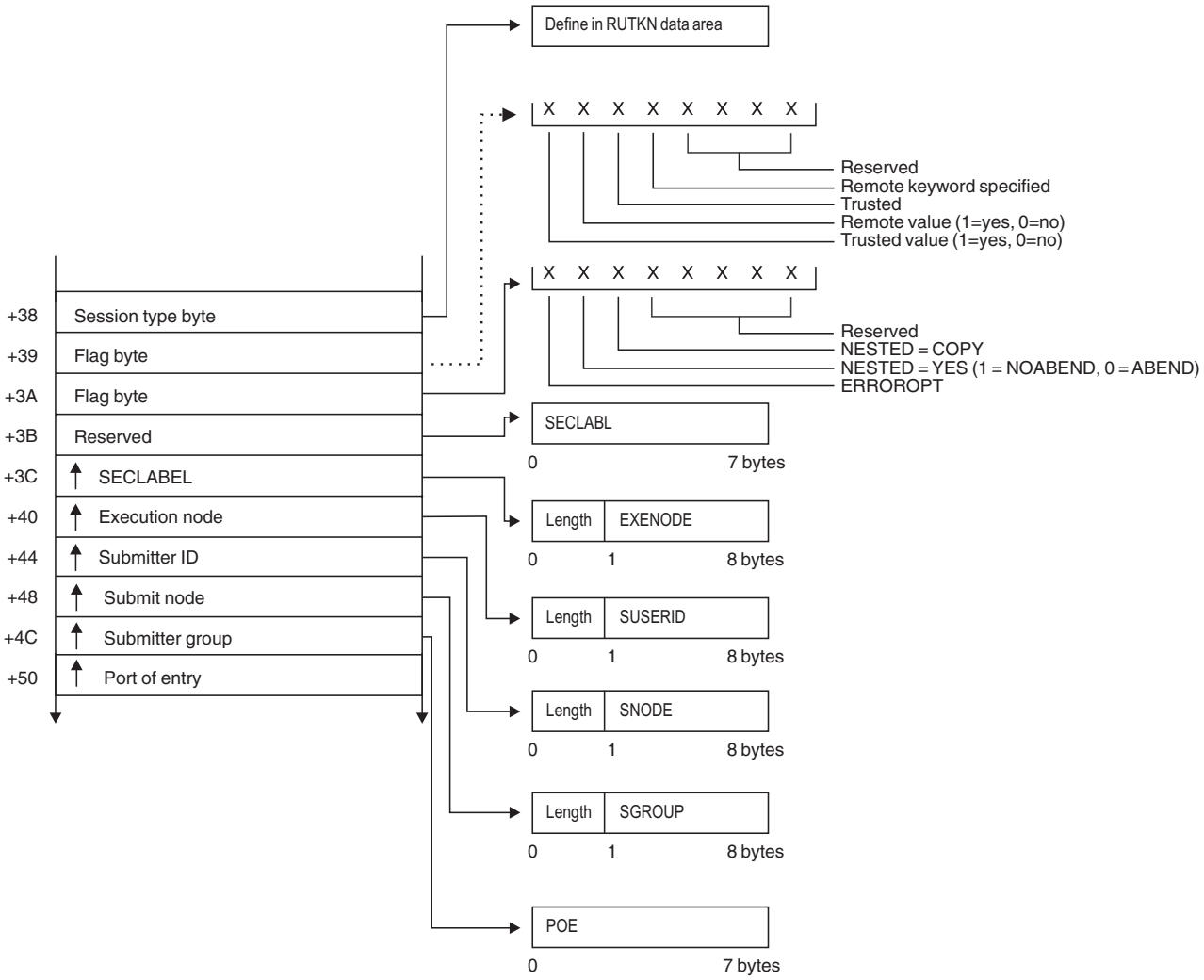


Figure 23. VERIFY/VERIFYX Service Parameter List (Part 3 of 4)

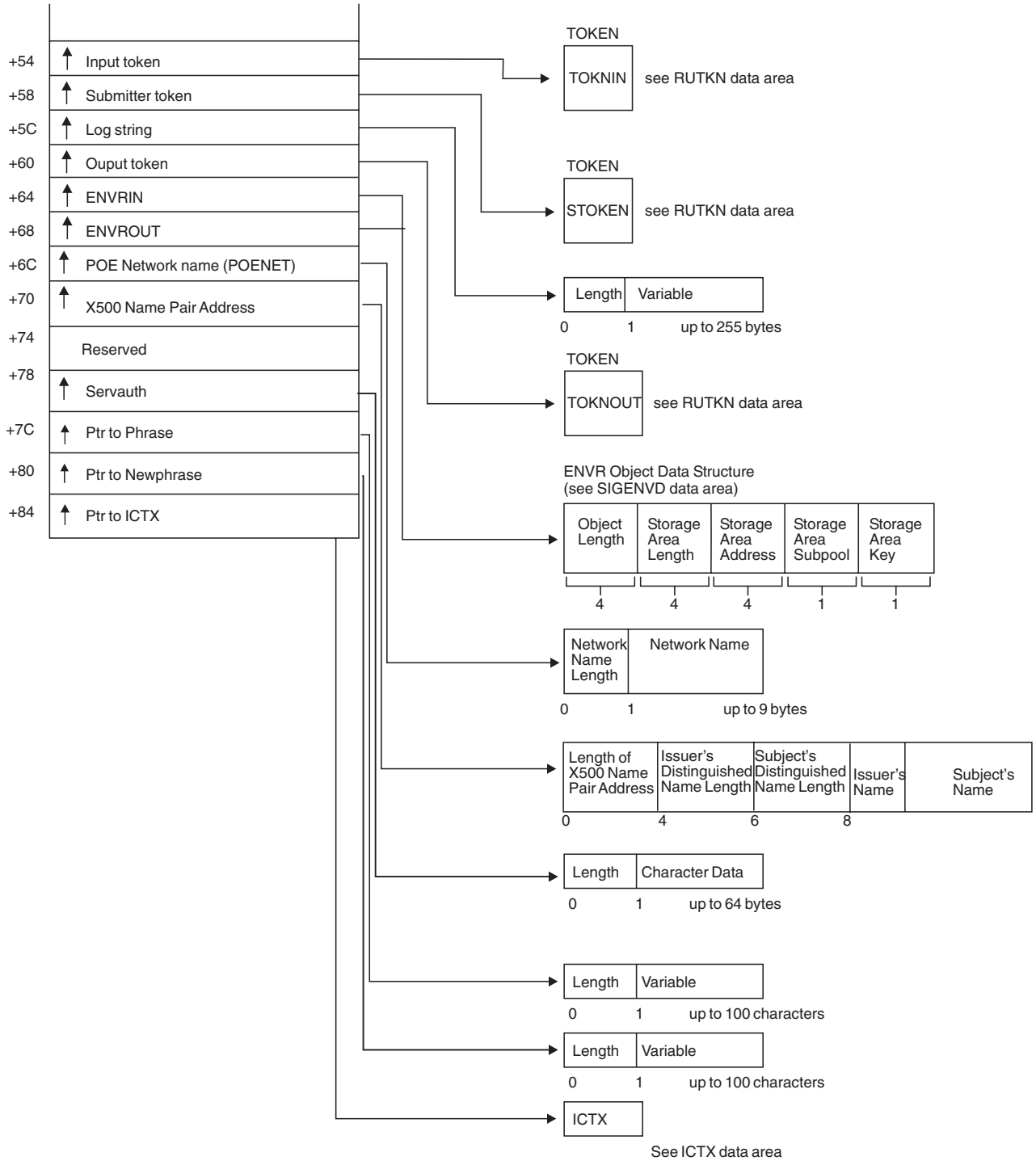


Figure 23. VERIFY/VERIFYX Service Parameter List (Part 4 of 4)

ICHEINTY parameter list

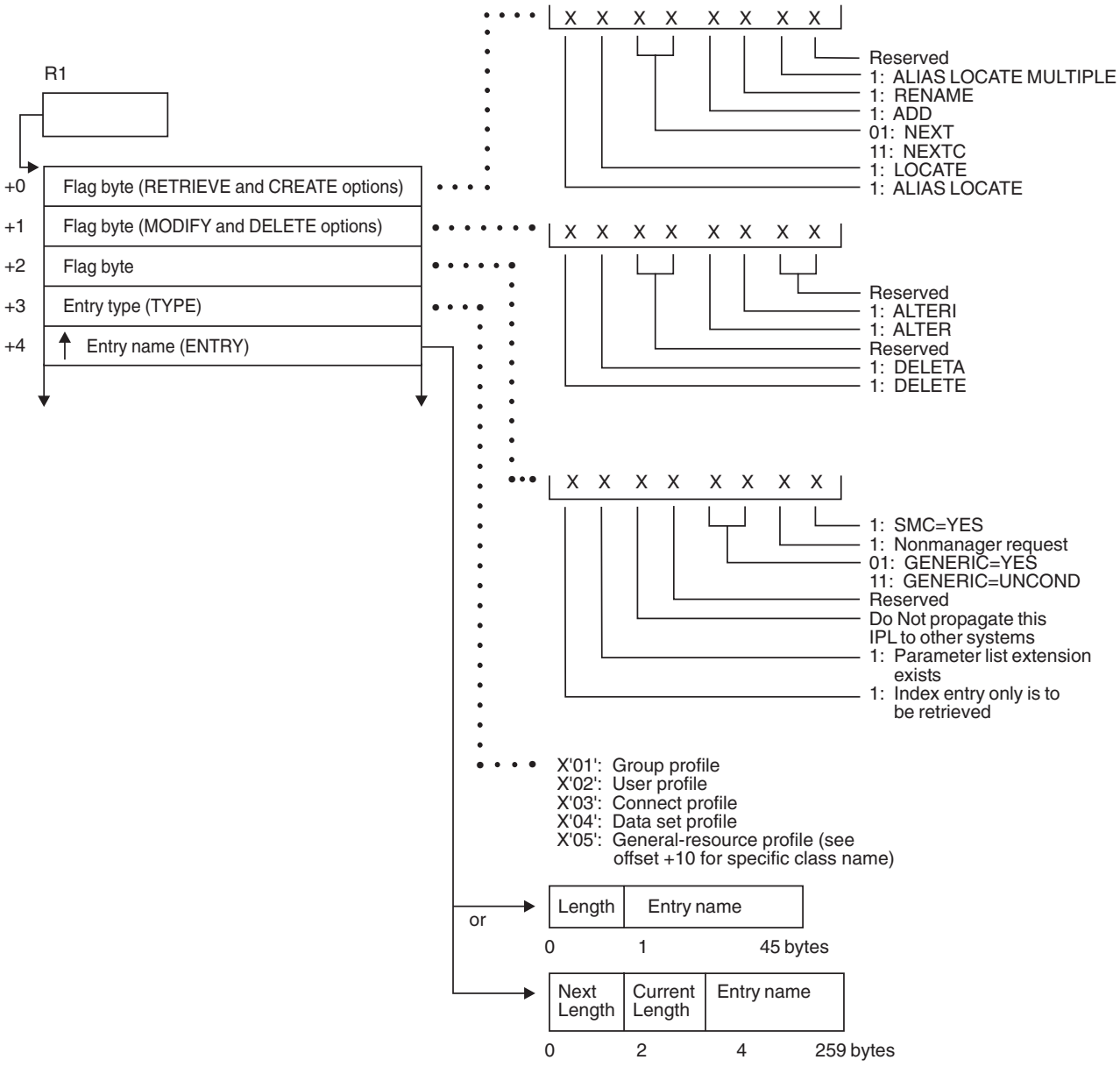
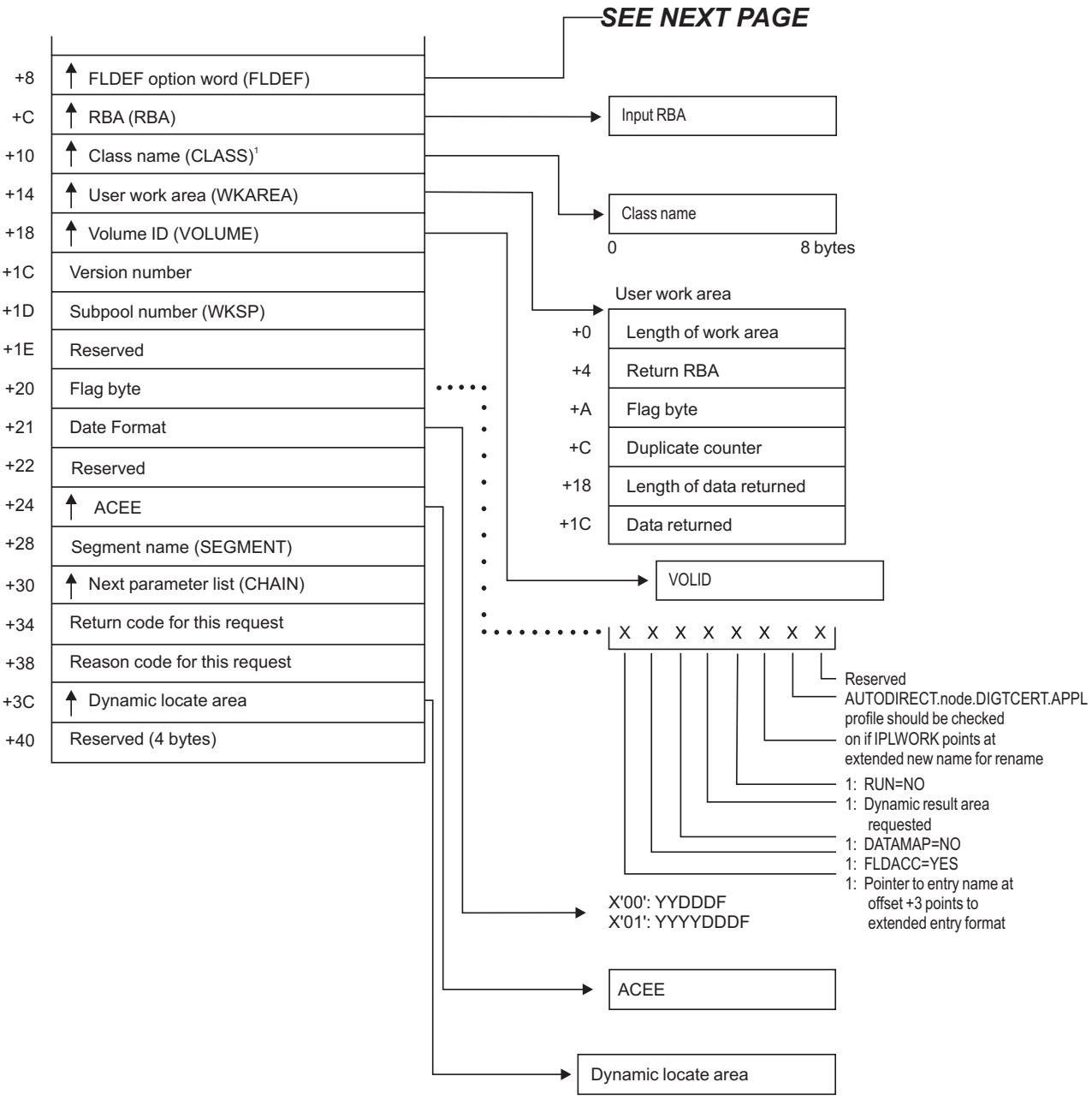


Figure 24. ICHEINTY Parameter List (Part 1 of 3)



[1] This field has a value only when the byte at +3 (entry type) has a value of X'05'.

Figure 24. ICHEINTY Parameter List (Part 2 of 3)

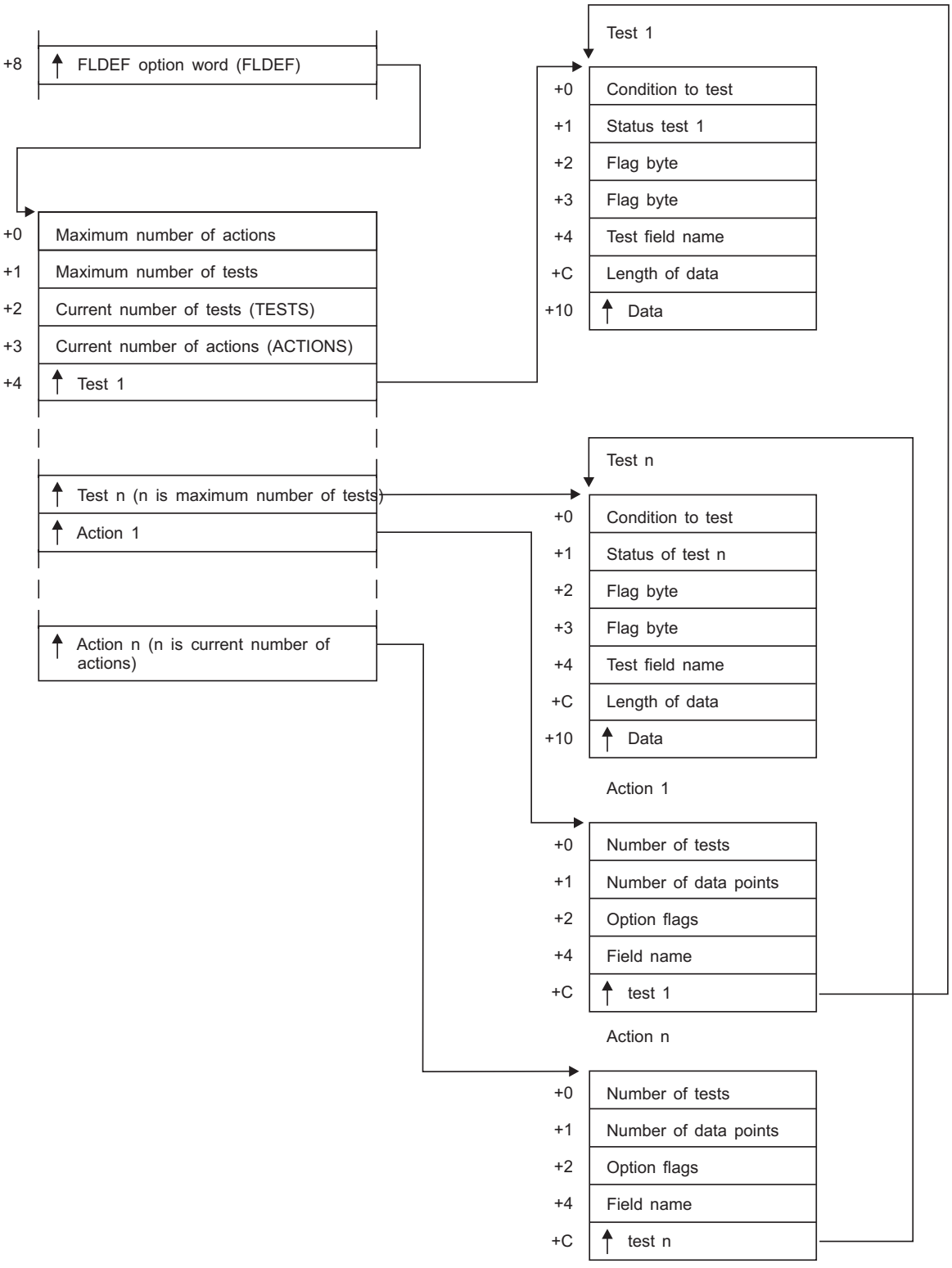


Figure 24. ICHEINTY Parameter List (Part 3 of 3)

Chapter 7. Diagnosis Reference for RACF on z/VM

The following sections contain information to help you diagnose problems running RACF on your z/VM system.

Process Flows for RACF on z/VM

On z/VM, RACF receives input from users in several ways:

- When the user enters RACF commands in a RACF command session. See "Process Flow for a RACF Command Session" below.
- When the user enters RACF commands using RAC. See "Process Flow for the RAC Command Processor" on page 122.
- When the user enters a CP command or DIAGNOSE function that requires RACF processing. See "Process Flows for CP Commands, DIAGNOSE Codes, and System Functions" on page 123.
- When the RACROUTE macro is invoked in an application program. See "Process Flow for RACROUTE Macro under CMS" on page 127.
- When an installation exit is invoked in the course of RACF processing. See "Process Flow for RACF Installation Exits on z/VM" on page 131.
- When the user invokes a RACF utility. See "Process Flow for RACF Utilities on z/VM" on page 132.

Process Flow for a RACF Command Session

A RACF command session is entered when a user enters the RACF command, and ends when the user enters the END command. During the session, the user can enter RACF commands such as LISTUSER, PERMIT, and others. Figure 25 illustrates the process flow for a RACF command session.

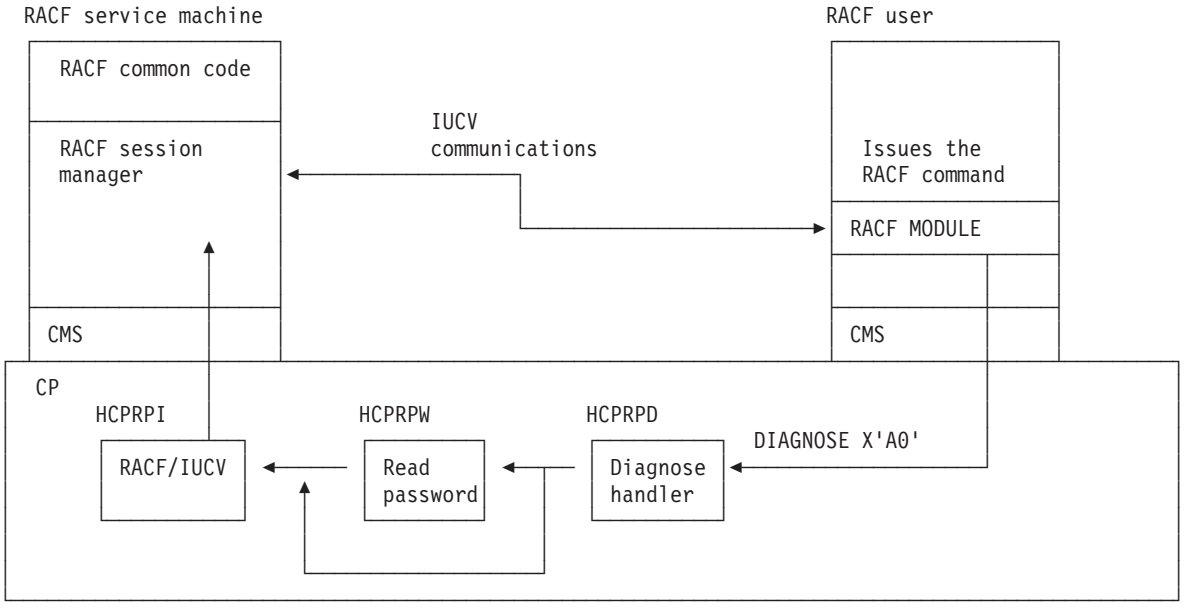


Figure 25. Process Flow for a RACF Command Session

1. Entering the RACF command from a CMS session causes the RACF MODULE to execute and puts the user into a RACF command session.

2. One of two actions occurs:
   - If password prompting is in effect, module HCPRPD calls module HCPRPW to prompt for the password and HCPRPW then passes the request to module HCPRPI.
   - If password prompting is not in effect, module HCPRPD passes the request directly to module HCPRPI.

   For more information on password prompting, see *Program Directory*.
3. After the password checking (if any), the RACF MODULE uses CMS IUCV (Inter-User Communication Vehicle) to communicate between the user's virtual machine and a RACF service machine.

   Once a RACF command session has been established, the user's machine has an IUCV path established with a RACF service machine.

   RACF commands entered during a RACF command session are handled by a RACF service machine. RACF simulates the TMP (terminal monitor program), which is part of the Time Sharing Option/Extensions [TSO/E] of z/OS, to parse RACF commands and to issue syntax error messages.
4. The user enters the END command to terminate the RACF command session.

Process Flow for the RAC Command Processor

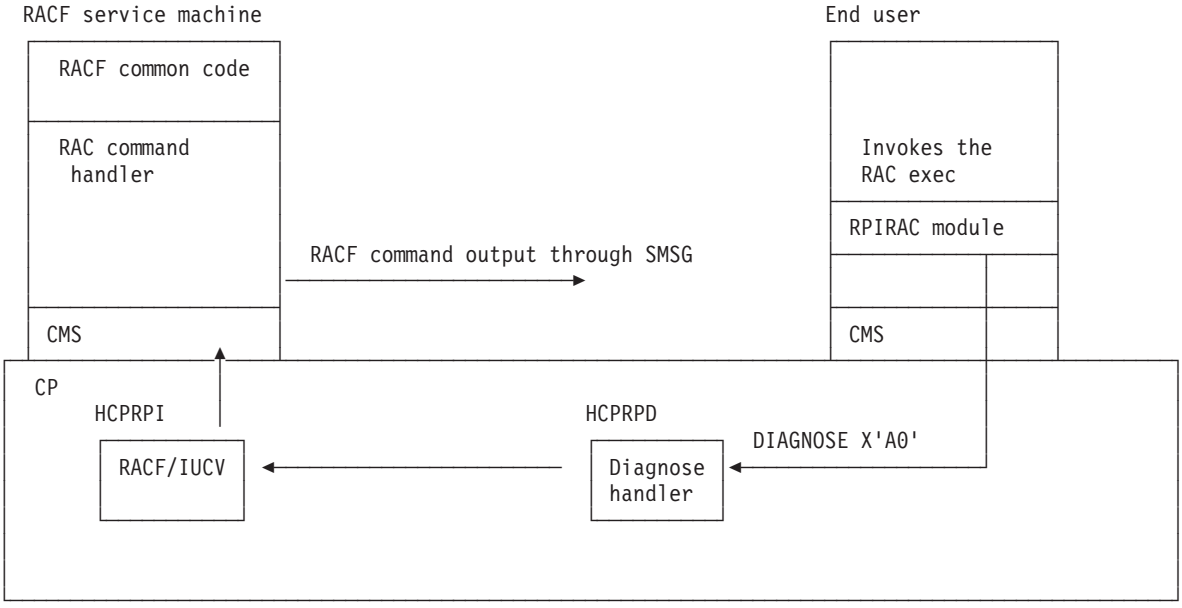


Figure 26. Process Flow for the RAC Command Processor

1. Invoking the RAC exec from a CMS session by entering RAC plus a RACF command (for example, RAC LU) causes the RPIRAC module to be invoked to process the command.
2. If the RACF command is one of the RACF SFS commands (ADDDIR, ADDFILE, ALTDIR, ALTFILE, DELDIR, DELFILE, LDIRECT, LFILE, PERMDIR, PERMFILE, SRDIR, SRFILE), RPIRAC first calls the ICHSFSDF exec. If the file pool ID, user ID, or both are omitted from an SFS format name, the ICHSFSDF exec inserts the current file pool ID, file space, or both into the SFS format name.

3. RPIRAC copies the command line and calls the RCMDRFMT exec. RPIRAC then issues Diagnose X'A0' subcode X'2C' passing the command line address in the RX register.
4. HCPRPD processes the request and passes the command line as a modified version of ACIPARMS to HCPRPI. HCPRPI sends the request to the appropriate RACF service machine for execution.
5. The RACF command is processed in a RACF service machine, and all terminal output generated is transmitted back to the end user through SMSG. The RPIRAC module running in the user virtual machine handles each SMSG from a RACF service machine by writing the buffer contents to the RACF DATA file. RPIRAC then calls the RACOUTP EXEC which displays the contents of the RACF DATA file to the user's console.
6. When all output from the RACF command terminates, the RPIRAC module returns control to CMS.

Process Flows for CP Commands, DIAGNOSE Codes, and System Functions

This section describes the process flow for the following:

- See Figure 27 for a description of the process flows for the LOGON, AUTOLOG, and XAUTOLOG commands.
- Use of any CP command, DIAGNOSE code, or system function when RACF is called to do auditing or access checking for the command. See Figure 28 on page 124.
- Use of the DIAGNOSE X'A0' function when RACF is called to do access checking for the request. See Figure 29 on page 125.

Figure 27 illustrates the process flow for the LOGON, AUTOLOG, and XAUTOLOG commands.

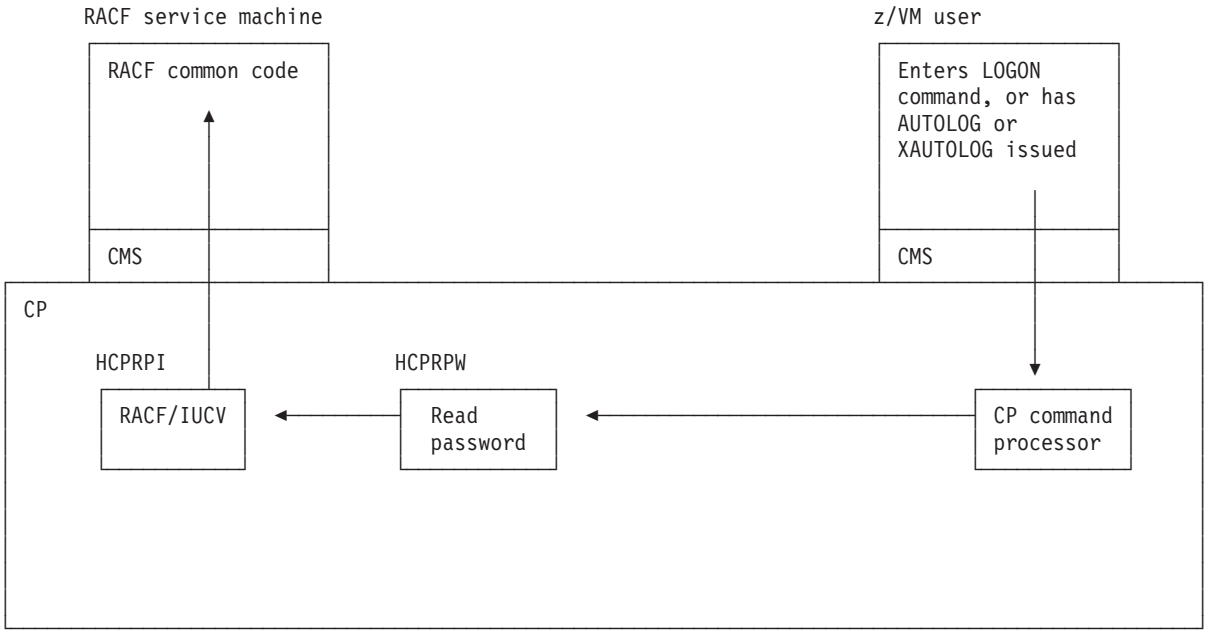


Figure 27. Process Flow for LOGON, AUTOLOG, and XAUTOLOG Commands

1. The user enters a LOGON command, or an AUTOLOG or XAUTOLOG command is issued to IPL a virtual machine.
2. The CP command processor passes ACIPARMS to HCPRPW. HCPRPW prompts for the password in the following two cases:
   - For LOGON, if the password or password phrase is omitted or is syntactically incorrect
   - For AUTOLOG and XAUTOLOG, if a password is required but is not supplied
3. HCPRPW calls HCPRPI.
4. Module HCPRPI sends the request to a RACF service machine.
5. RACF processes the request and returns a return code to the CP command processor. The CP command processor then either allows or disallows the user request based on the return code.

Figure 28 illustrates the process flow for certain CP commands, DIAGNOSE codes, and system functions.

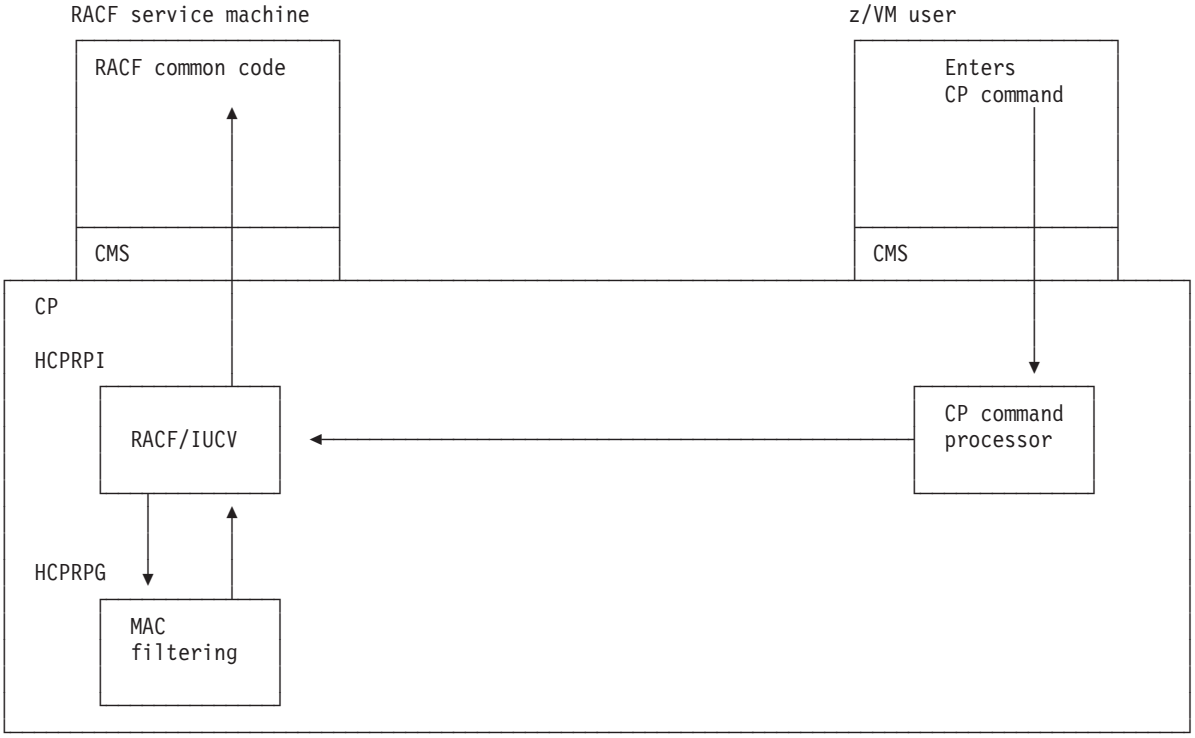


Figure 28. Process Flow for Certain CP Commands, DIAGNOSE Codes, and System Functions

1. The user enters a CP command.
2. The CP command processor for the command calls module HCPRPI.
3. Module HCPRPI sends the request to module HCPRPG to determine whether the server can be bypassed for an event protected by the VMMAC class.
   - If HCPRPG finds a match on the user/resource SECLABEL pairing, the request is returned to the command processor.
   - If a match is not found, module HCPRPI sends the request to a RACF service machine.
4. RACF processes the request. If auditing is done for the CP command, RACF generates an SMF record. If access checking is done for the CP command,

RACF returns a return code to the CP command processor. In addition, if SECLABEL checking is done for the command, RACF indicates whether or not the MAC filter should be updated with this SECLABEL pairing.

5. HCPRPI regains control when the service machine completes the request. If the MAC filter needs to be updated with a SECLABEL pairing, then HCPRPI calls HCPRPG to do so. HCPRPI then passes control back to the CP command processor.
6. The CP command processor then either allows or disallows the user request based on the return code.

Figure 29 illustrates the process flow for DIAGNOSE X'A0'.

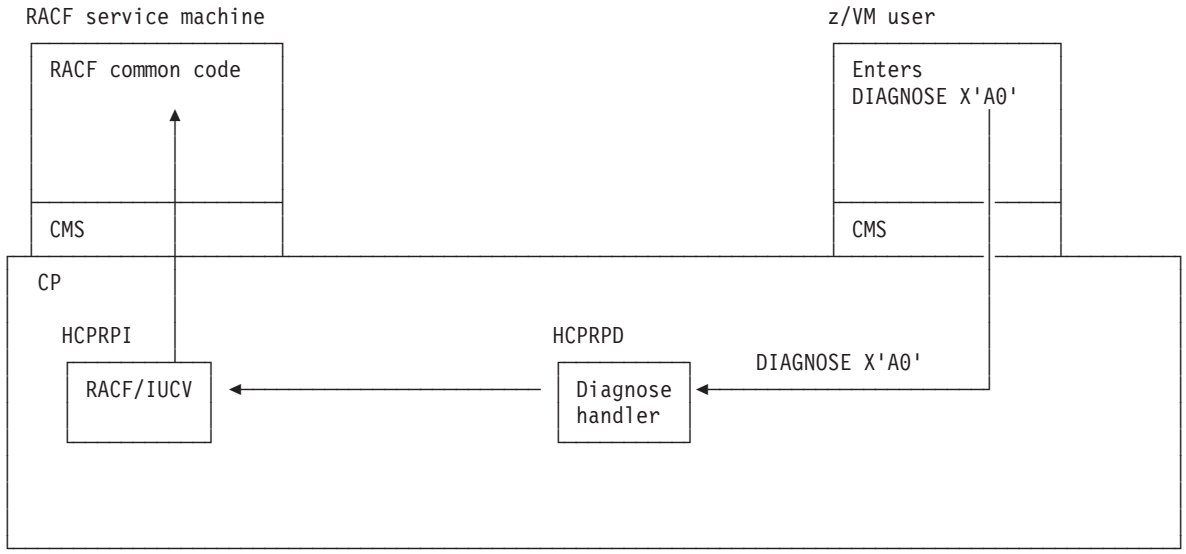


Figure 29. Process Flow for DIAGNOSE X'A0'

1. The user enters a DIAGNOSE X'A0' function.
2. The DIAGNOSE function processor (HCPRPD) calls module HCPRPI.
3. Module HCPRPI sends the request to a RACF service machine.
4. RACF processes the request and returns a return code to the DIAGNOSE processor. The DIAGNOSE processor then either allows or disallows the user request based on the return code.

Explanation of DIAGNOSE X'A0' Subcodes

Table 19. List of the Diagnose X'A0' subcodes

Subcode	Description
X'00'	Retrieve groupname for a given user ID
X'04'	Verify the user ID and validate the password for that user ID
X'08'	Indicate if access control interface is installed
X'0C'	Perform MAC functions
X'10'	Audit processing
X'14'	Request connection to the RACF command session
X'18'	Activate/Deactivate RACF
X'1C'	Register RACF to z/VM as a POSIX-compliant ESM

Table 19. List of the Diagnose X'A0' subcodes (continued)

Subcode	Description
X'20'	Limited function RACROUTE
X'24'	Fetch security related data
X'28'	Update security related data
X'2C'	Allow for RAC command processor
X'30'	Query a user's current SECLABEL
X'34'	Update the human-readable-label to SECLABEL correlation table ("HR table") in CP.
X'38'	Obtain the size of, or a copy of the human-readable-label to SECLABEL correlation table ("HR table").
X'3C'	Verify the user ID and validate the password phrase for that user ID.

You can use RACF profiles in the VMCMD class to protect four of these subcodes (X'04', X'30', X'34', and X'3C'). For additional information about these subcodes and the return codes that can be generated as a result of protecting them, refer to *z/VM: RACF Security Server Macros and Interfaces*.

Subcode X'38' is also a programming interface but cannot be protected by a VMCMD profile. For information about subcode X'38', see *z/VM: RACF Security Server Macros and Interfaces*.

Subcodes X'00' and X'08' are provided with z/VM and are documented in *VM/ESA CP Programming Services*. The other subcodes are not programming interfaces, but are used within the RACF product. They are listed in this section for diagnostic purposes only.

Process Flow for RACF Macros in the RACF Service Machine

The RACF service machine simulates a z/OS environment in which the RACROUTE macro, and the independent system macros (e.g. RACINIT, RACHECK, etc.) execute when issued within the service machine itself. These macros could be issued by the RACF product itself, or from within customer exit routines. The following figure diagrams the process flow in this environment.

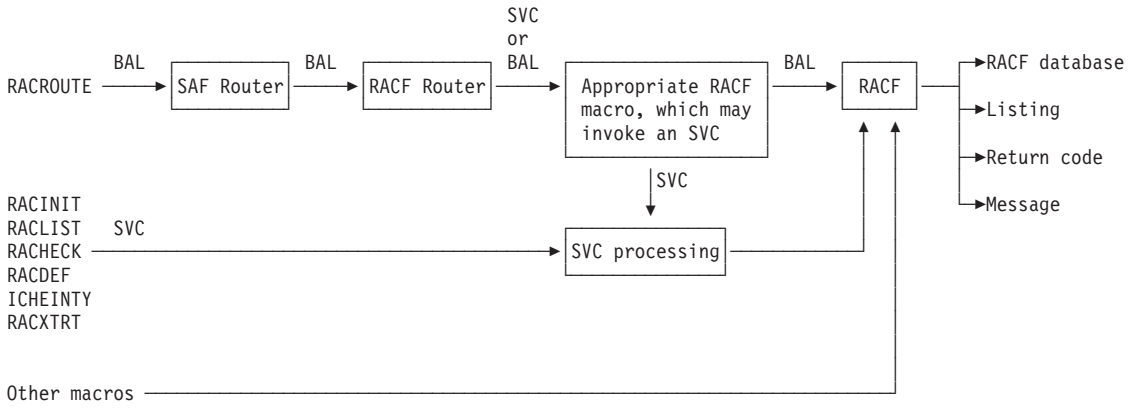


Figure 30. Process Flow for RACF Macros on z/OS

Process Flow for RACROUTE Macro under CMS

This process flow applies only to the RACROUTE macro when invoked in an application program running on a virtual machine other than a RACF service machine).

Figure 31 on page 128 illustrates the process flow for the RACROUTE macro on z/VM under CMS.

Figure 32 on page 130 illustrates the process flow for the RACROUTE macro on z/VM under GCS.

Note: RACF for z/VM supports both synchronous and asynchronous mechanisms for notifying the resource manager when a RACROUTE request has been completed. The process flows that follow describe the synchronous mechanism. For more information on ECBs and their use in the asynchronous mechanism, see *z/VM: RACF Security Server Macros and Interfaces*.

Process Flow for the RACROUTE Macro on z/VM with CMS

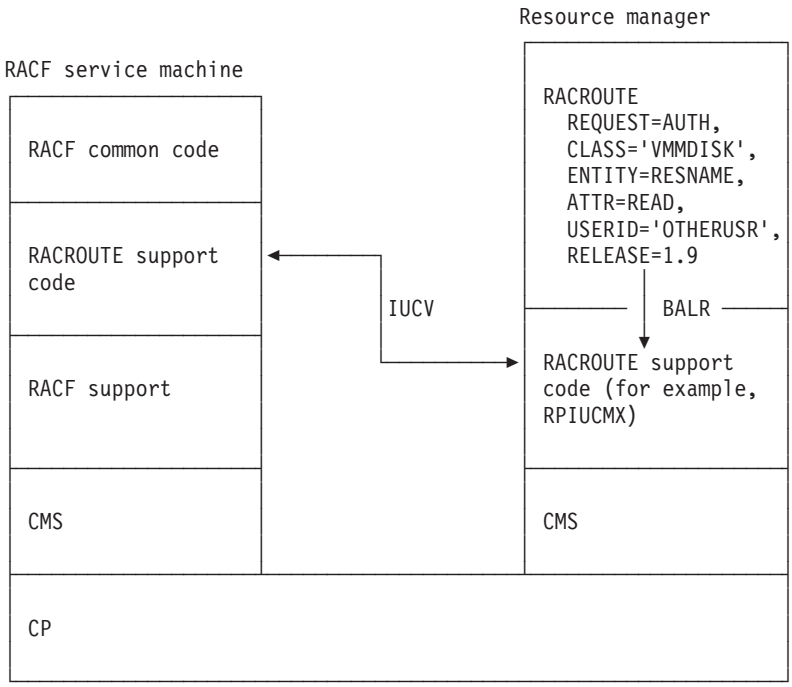


Figure 31. Process Flow for the RACROUTE Macro on z/VM with CMS

Initialization of the RACROUTE Interface

Before initialization of the RACROUTE interface can begin, the following requirements must be met:

- An IUCV card must be in the CP directory.
- The resource manager must be permitted to the ICHCONN profile in the FACILITY class with READ or UPDATE access. See *z/VM: Security Server RACROUTE Macro Reference* for information on which request type requires which access.

Initialization consists of the following steps.

1. The resource manager issues `RPIUCMS INIT`.
2. Module RPIUCMS does a NUCXLOAD of RPIUCMX
3. Module RPIUCMX attempts to connect to a RACF service machine by reading the ID in the RACF SERVMACH file and connecting to that ID.
4. The SFSAUTOACCESS value is also read from the RACF SERVMACH file. It is kept for the duration of the IUCV session.
5. Once connection is accepted by a RACF service machine, module RPIUCMX sets up a simulated vector table that contains the entry points into module RPIUCMX used in RACROUTE macro invocations.

Request Processing

1. An application program invokes the RACROUTE macro, which calls module RPIUCMX.
2. SFSAUTOACCESS checking is done to determine if access will be granted without sending the request to a RACF service machine.

3. If a preprocessing exit address has been specified on the RACROUTE invocation, that exit is called.
4. The request is sent to a RACF service machine.
5. Within that RACF service machine, the request is processed. The updated return and reason codes, along with any data, are returned to the resource manager machine.
6. In the resource manager machine, module RPIUCMS receives control from CMS and updates the original parameter list.If a postprocessing exit address has been specified on the RACROUTE invocation, that exit is called. If the exit indicates processing should continue, control is returned to module RPIUCMX, then to CMS, then to the resource manager. When the resource manager continues, the return and reason codes appropriate to the RACROUTE request have been sent.

Termination

1. The resource manager issues `RPIUCMS TERM`.
2. Work area storage is freed, and the path is severed.
3. Validation logic and the class descriptor table (CDT) are purged from virtual storage, and RPIUCMX is dropped as a nucleus extension.

Process Flow for the RACROUTE Macro on z/VM with GCS

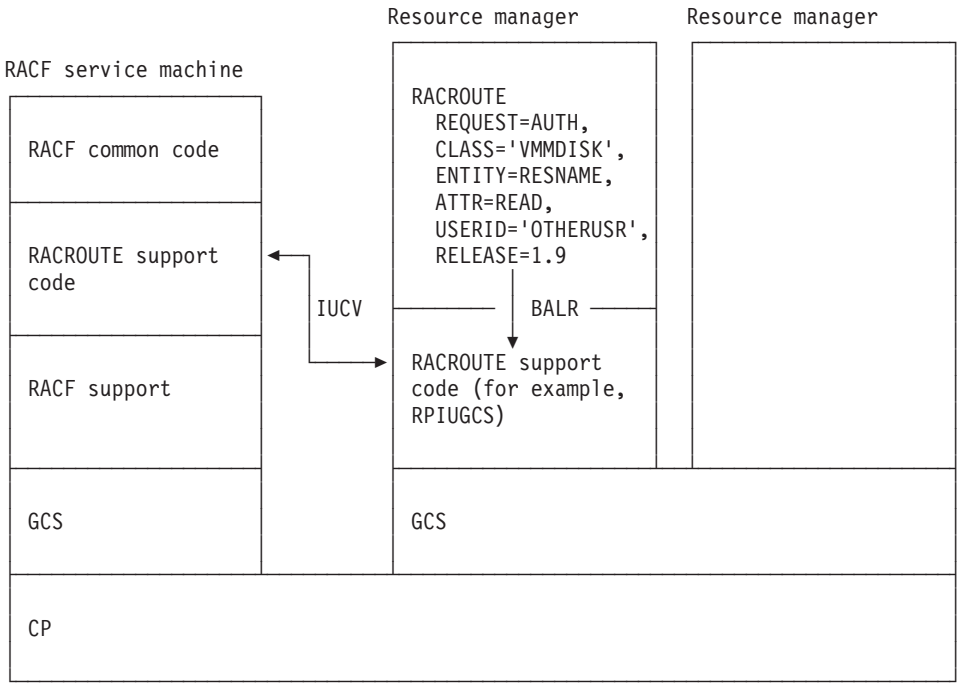


Figure 32. Process Flow for the RACROUTE Macro on z/VM with GCS

Initialization of the RACROUTE Interface

Before initialization of the RACROUTE interface can begin, the following requirements must be met:

- An IUCV card must be in the CP directory.
- The resource manager must be permitted to the ICHCONN profile in the FACILITY class. With READ or UPDATE access, see *z/VM: Security Server RACROUTE Macro Reference* for information on which request type requires which access.
- The user must have access to supervisor state and be authorized to use GCS functions. For more information, see the *VM/ESA Group Control System* or *z/VM Group Control System* books and *z/VM: Security Server RACROUTE Macro Reference*

Initialization consists of the following steps.

1. The resource manager issues
   GLOBAL LOADLIB RPIGCS
   LOADC RPIUGCS RPIATGCS
   RPIUGCS INIT
2. RPIATGCS attaches RPIUGCS.
3. Module RPIUGCS is loaded into storage from the RPIGCS LOADLIB on the GCS Y disk.
4. Module RPIUGCS attempts to connect to a RACF service machine by reading the ID in the RACF SERVMACH file and connecting to that ID.
5. Once connection is accepted by a RACF service machine, module RPIUGCS sets up a simulated vector table that contains the entry points into module RPIUGCS used in RACROUTE macro invocations.

Request Processing

1. An application program invokes the RACROUTE macro, which calls module RPIUGCS.
2. If a preprocessing exit address has been specified on the RACROUTE invocation, that exit is called.
3. Module RPIURIM sends the request to a RACF service machine.
4. Within that RACF service machine, the request is processed. The updated return and reason codes, along with any data, are returned to the resource manager machine.
5. In the resource manager machine, module RPIUGCS receives control from GCS and updates the original parameter list. If a postprocessing exit address has been specified on the RACROUTE invocation, that exit is called. If the exit indicates processing should continue, control is returned to module RPIUGCS, then to GCS, then to the resource manager. When the resource manager continues, the return and reason codes appropriate to the RACROUTE request have been sent.

Termination

1. The resource manager issues `RPIUGCS TERM`.
2. Work area storage is freed, and the path is severed.
3. Validation logic and the class descriptor table (CDT) are purged from virtual storage, and RPIUGCS is dropped as a nucleus extension.

Process Flow for RACF Installation Exits on z/VM

Except for the RACROUTE macro, RACF macros are allowed only in RACF user exits and run in the primary RACF service machine.

Figure 33 on page 132 illustrates the process flow for RACF installation exits on z/VM.

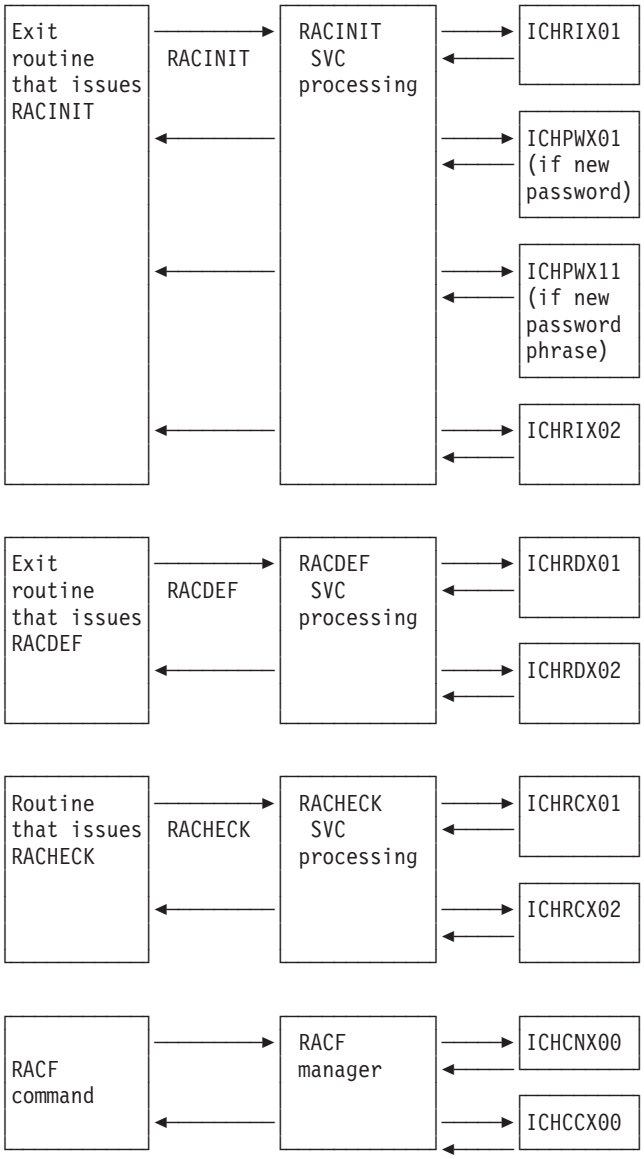


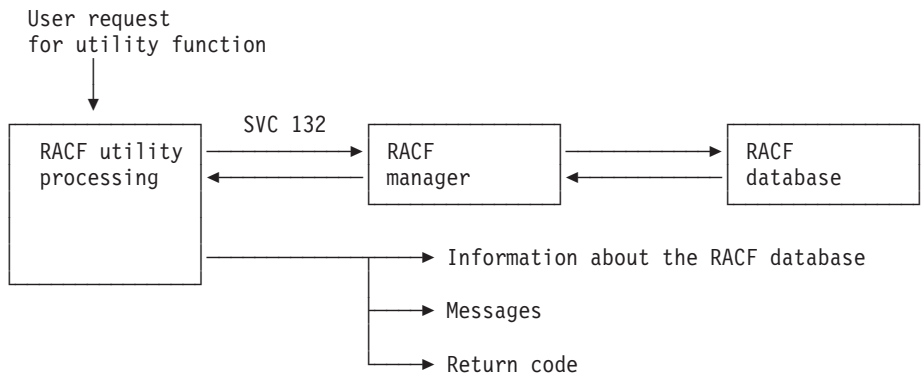
Figure 33. Process Flow for RACF Installation Exits on z/VM

Process Flow for RACF Utilities on z/VM

General-purpose RACF EXECs (for example, RACFPERM, RACFLIST) run in the general-user machine. They issue RACF commands using either the RAC command or the RACF command session. See “Process Flow for the RAC Command Processor” on page 122 and “Process Flow for a RACF Command Session” on page 121 for more information.

Utilities used to perform administrative functions which affect the RACF database (for example: IRRUT100, IRRUT200) generally run on a user ID with links to a RACF service machine’s minidisks. When one of these utilities runs, the RACF utility processing is as follows :

RACF Cross-Reference Utility
(IRRUT100 for RACF database
and **RACF Database Unload Utility**
(IRRDBU00):



Other RACF Utilities:

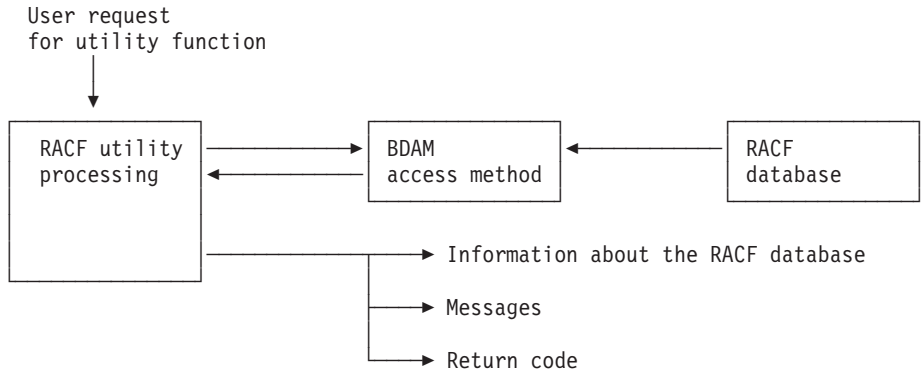


Figure 34. Process Flow for RACF Utilities

Process Flow for the RACF Report Writer on z/VM

Figure 35 on page 134 illustrates the process flow for RACF Report Writer on z/VM.

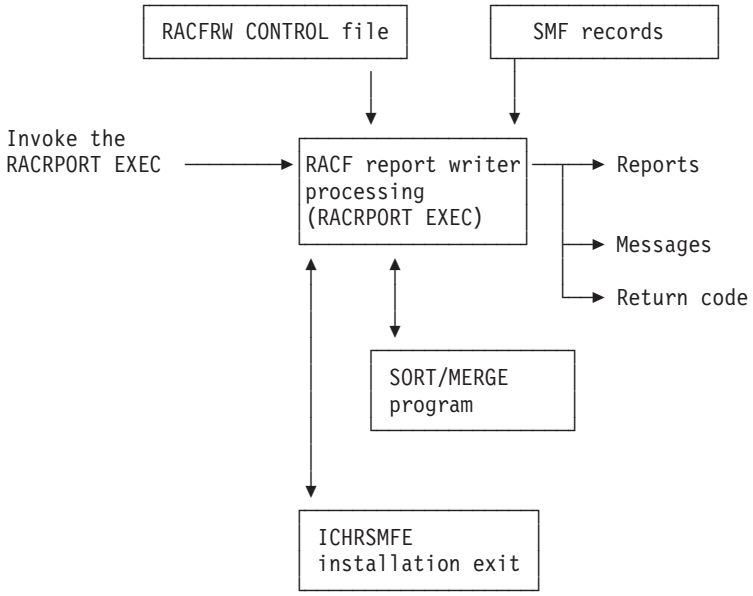


Figure 35. Process Flow for RACF Report Writer on z/VM

Control Block Overview for RACF on z/VM

This section shows the linkage between z/VM control blocks that are used for RACF.

Figure 36 illustrates the control blocks useful for diagnosing RACF on z/VM.

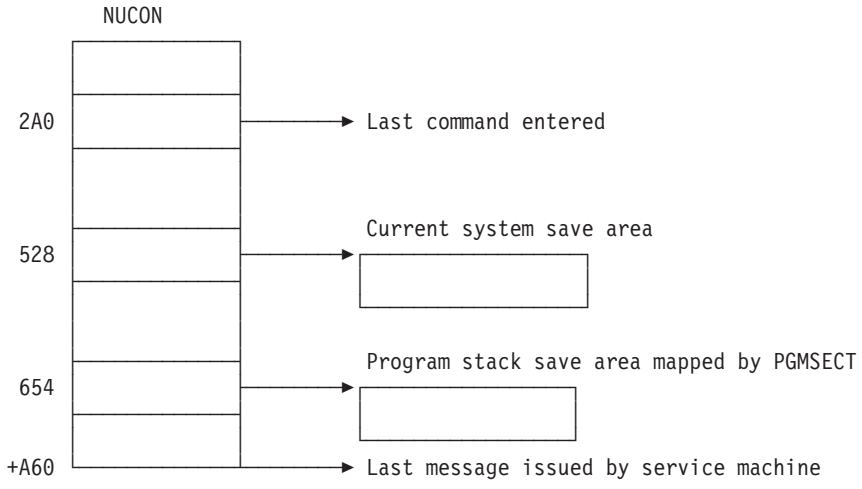


Figure 36. Control Blocks Useful for Diagnosing RACF on z/VM

Figure 37 on page 135 illustrates the control block overview for the RCVT and SAF router vector tables.

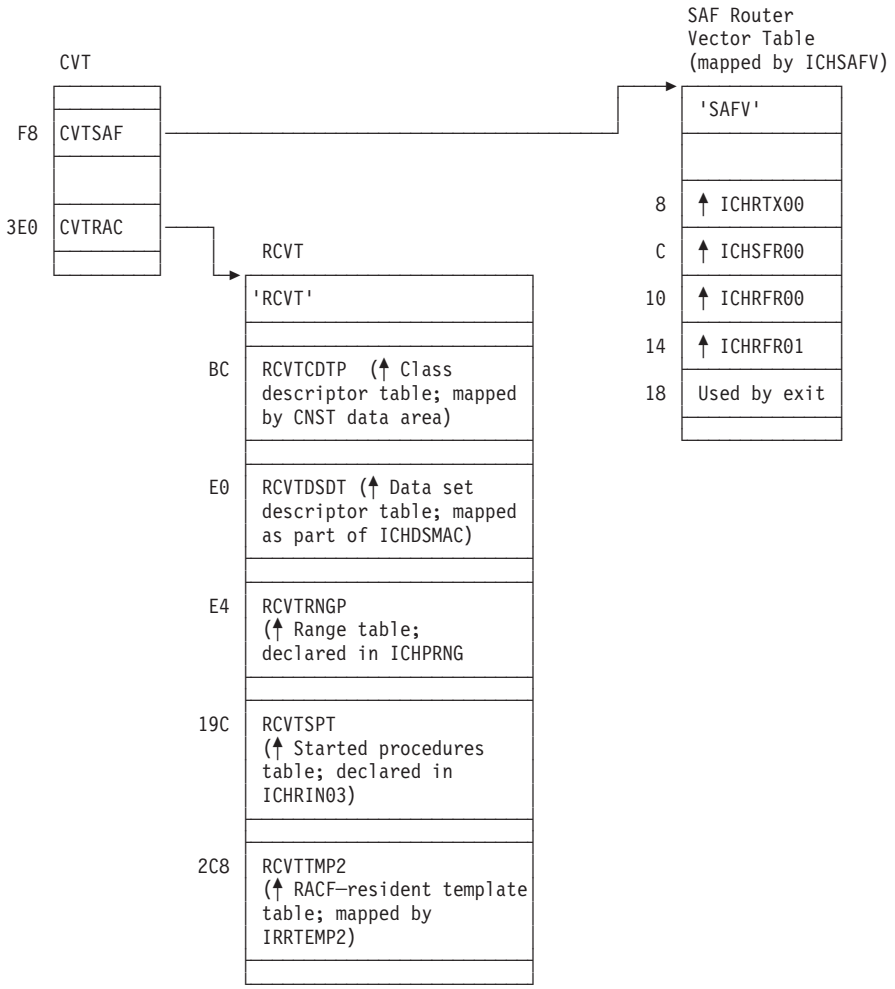


Figure 37. Control Block Overview: RCVT and SAF Router Vector Table

Note: For the mapping of the RCVT data area, see *z/VM: Security Server RACROUTE Macro Reference*.

On z/VM, when a RACINIT macro is issued, there is only one ACEE, and it is pointed to by the ASXB (as in Figure 38 on page 136).

For more information on how ACEEs are created and which ACEE is used, see *z/VM: RACF Security Server System Programmer's Guide*.

Figure 38 on page 136 illustrates the control block overview for ACEE on z/VM.

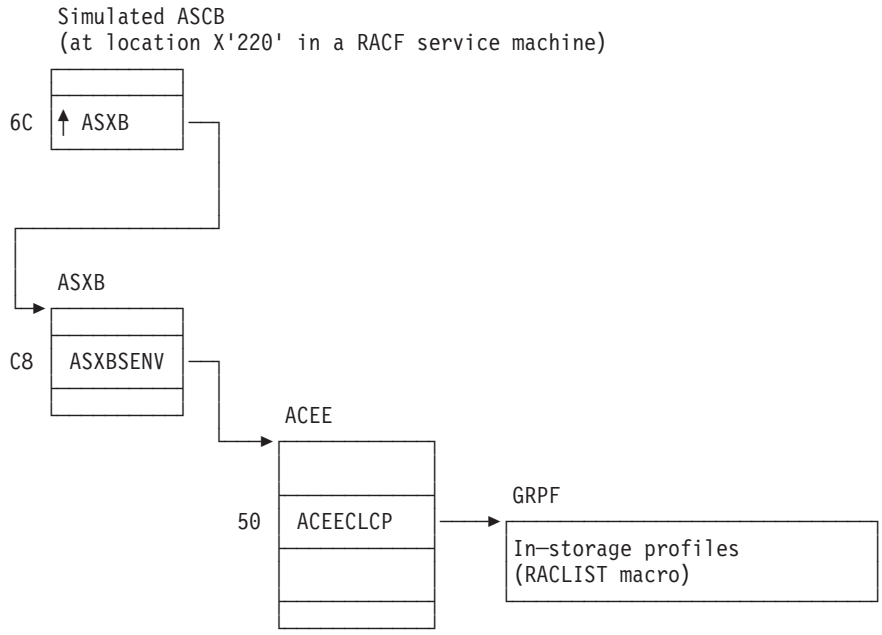


Figure 38. Control Block Overview: ACEE on z/VM

Note: For the mapping of the ACEE data area, see *z/VM: Security Server RACROUTE Macro Reference*.

Problem Determination for a RACF Service Machine

The following problem determination flowchart (Figure 39 on page 137) can be used whenever a RACF service machine fails to initialize. This could occur during, but is not limited to, the following times:

- When RACF is installed on z/VM
- When the system is IPLed
- When a RACF service machine is IPLed.

If a RACF service machine fails to initialize, follow the flowchart to isolate the cause of the problem and to find the suggested solution.

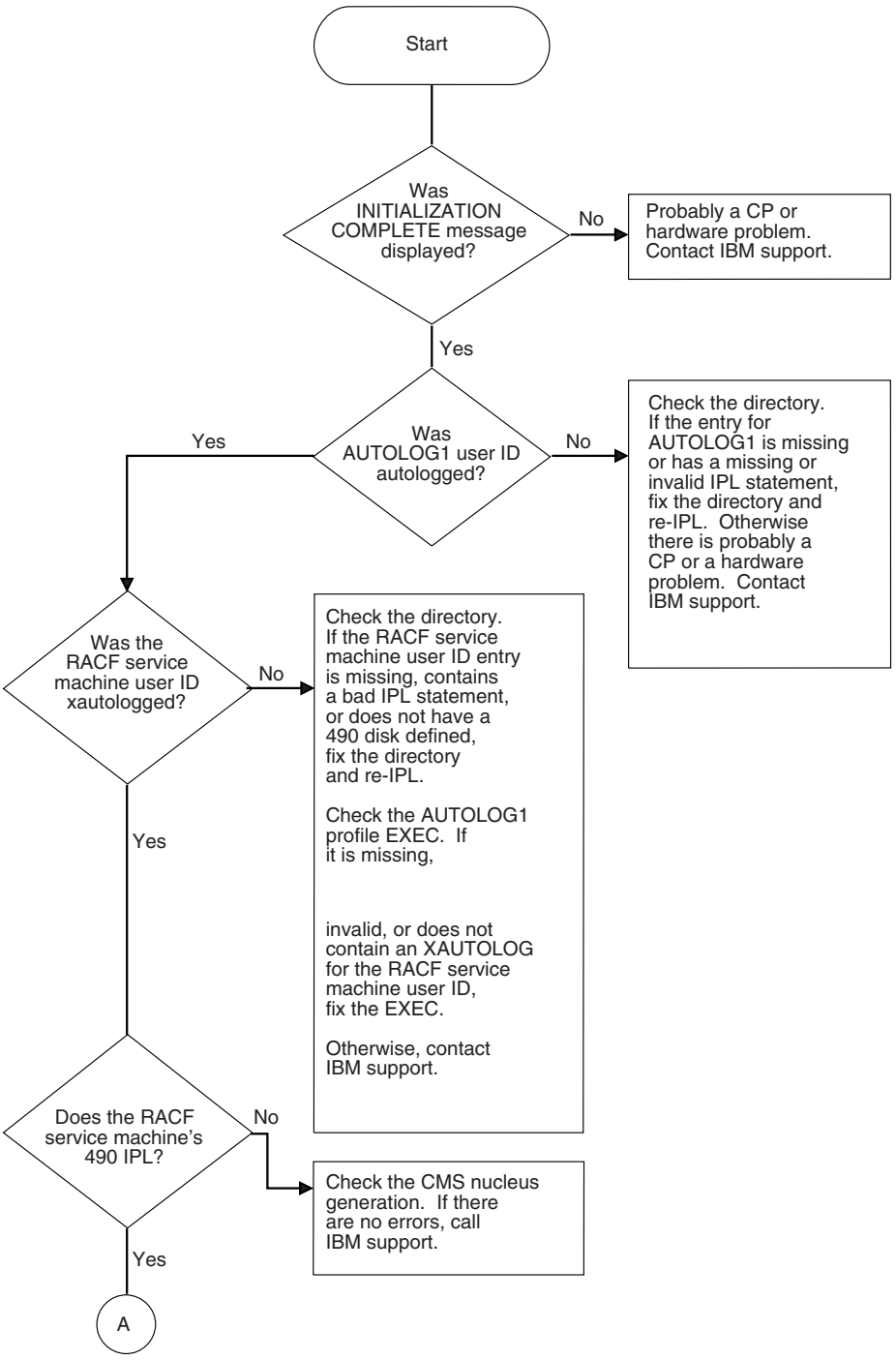


Figure 39. Problem Determination Chart for RACF z/VM Initialization (Part 1 of 6)

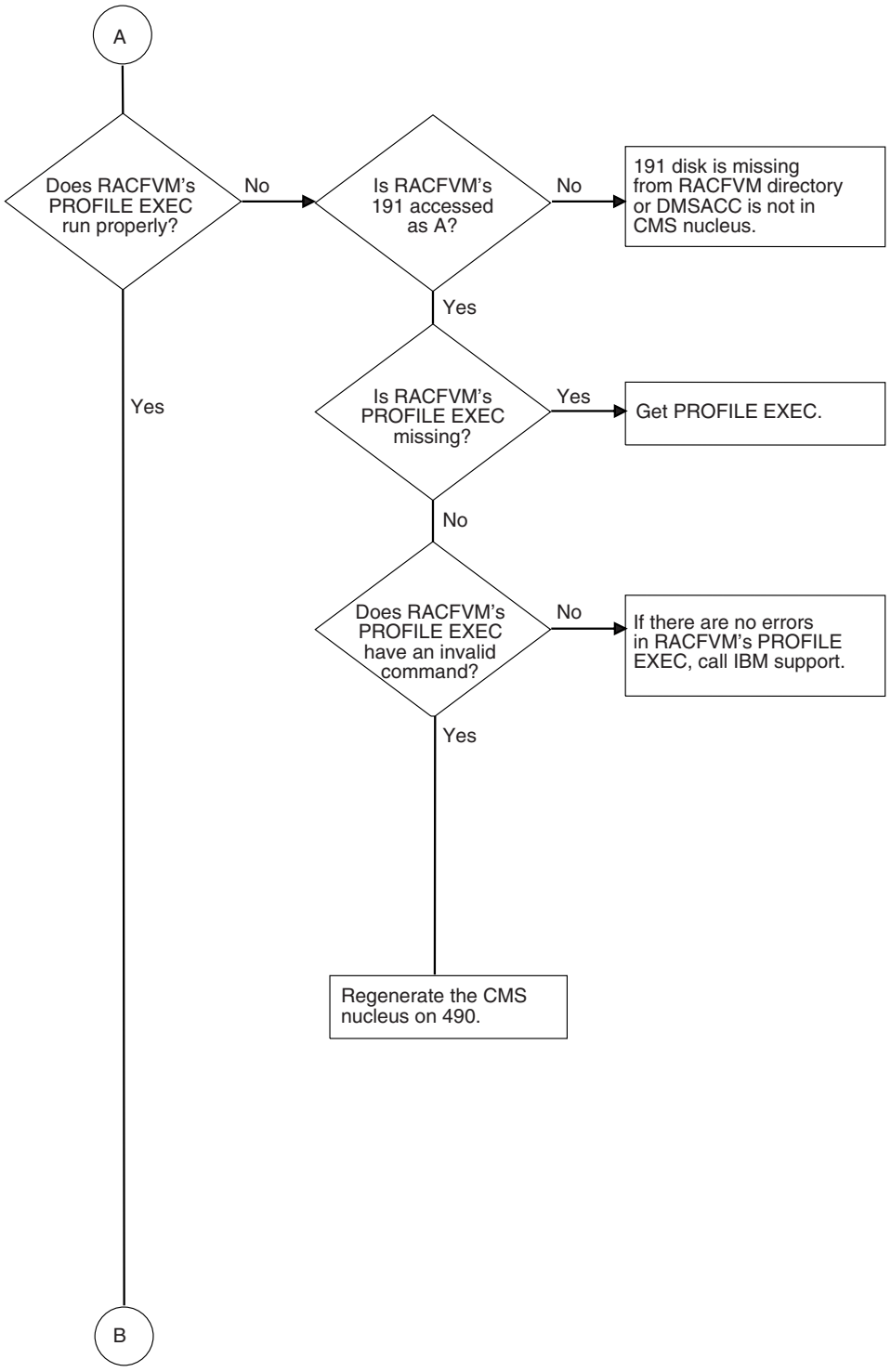


Figure 39. Problem Determination Chart for RACF z/VM Initialization (Part 2 of 6)

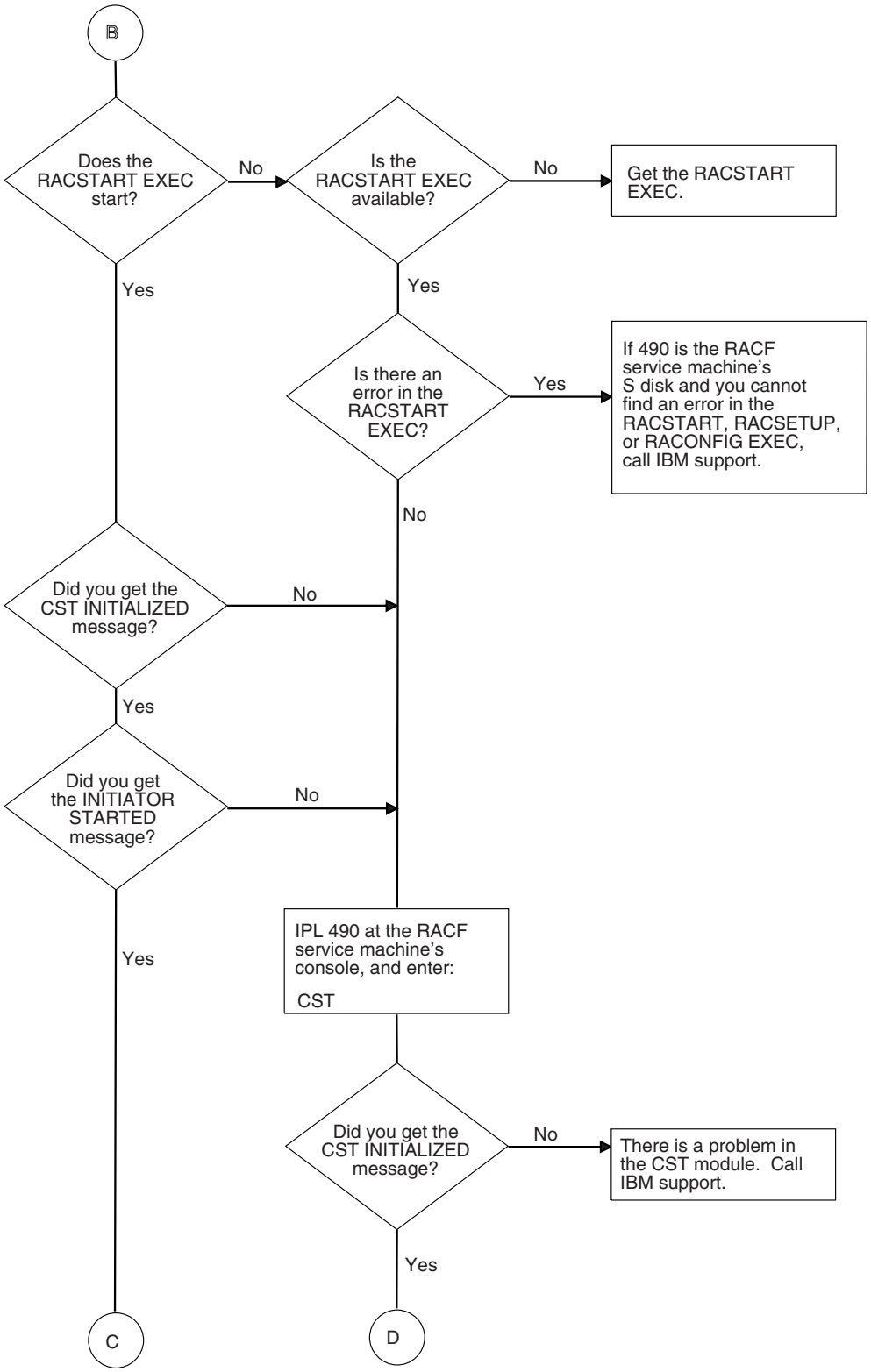


Figure 39. Problem Determination Chart for RACF z/VM Initialization (Part 3 of 6)

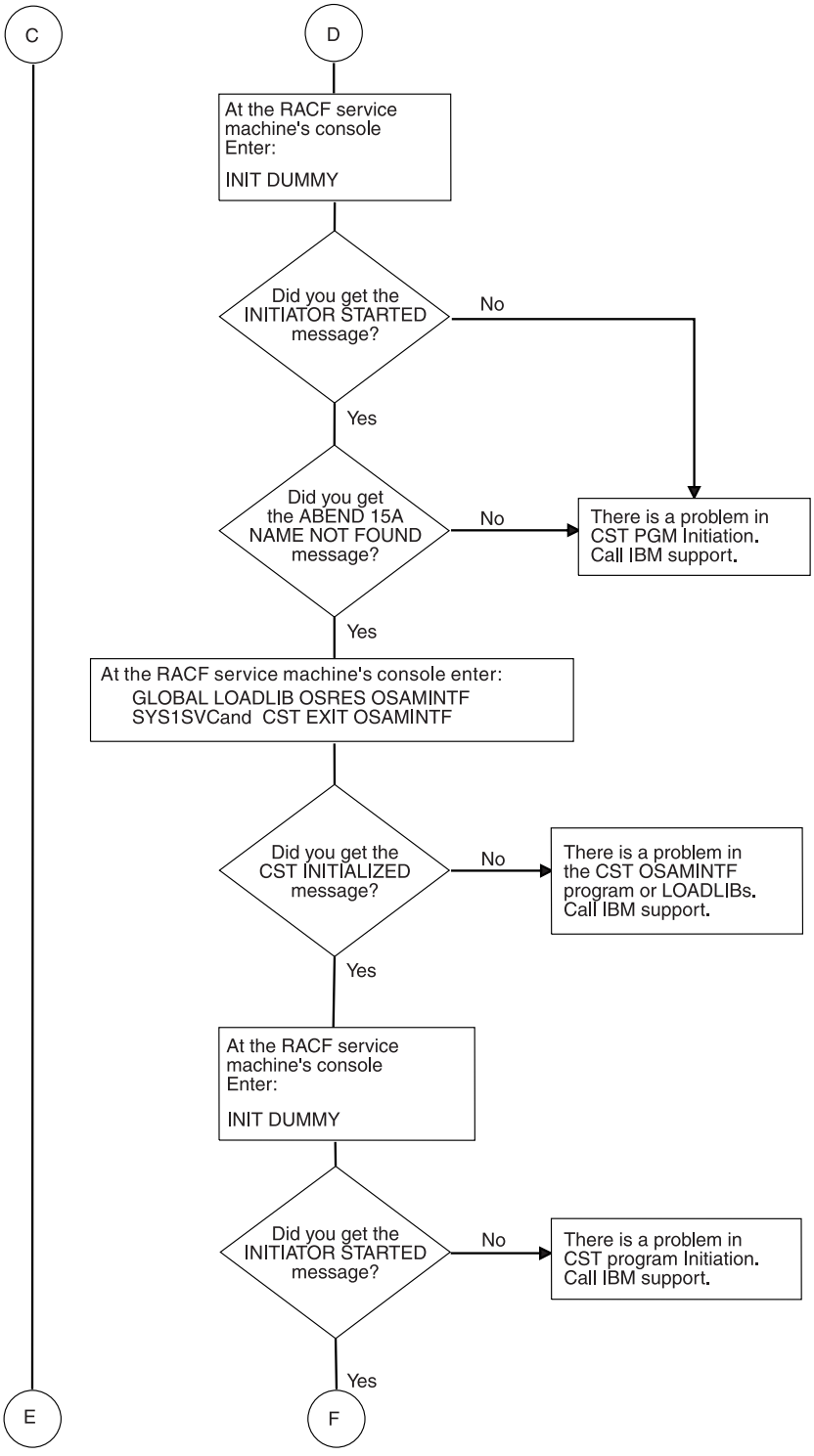


Figure 39. Problem Determination Chart for RACF z/VM Initialization (Part 4 of 6)

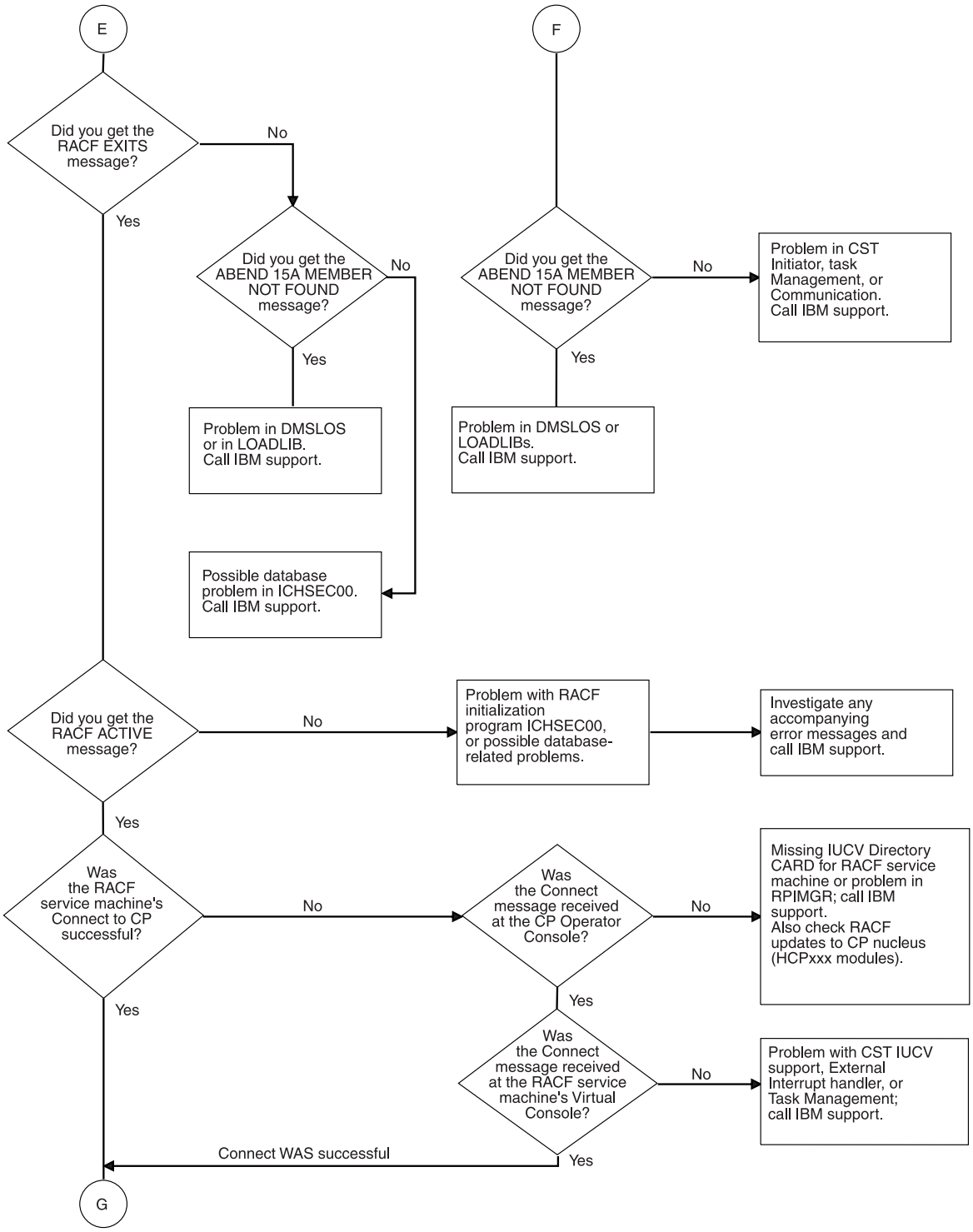


Figure 39. Problem Determination Chart for RACF z/VM Initialization (Part 5 of 6)

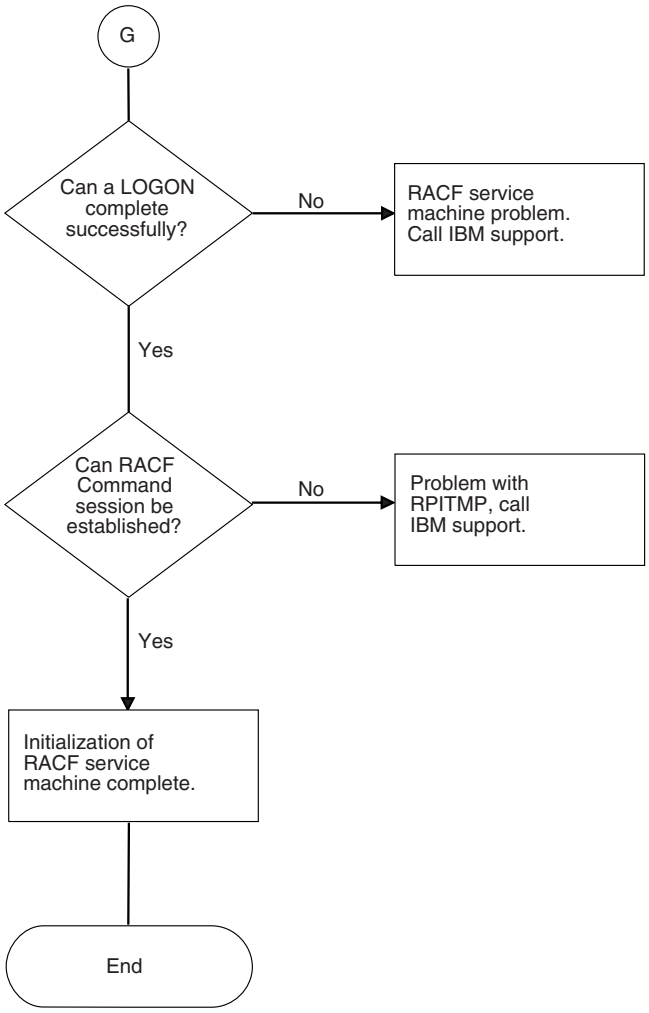


Figure 39. Problem Determination Chart for RACF z/VM Initialization (Part 6 of 6)

Part 3. Appendixes

Appendix. The ICB Data Area

The data area, ICB, is included here to help with the task of diagnosis on RACF.

ICB

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: Key: Residency:
Size:	1024 or 4096 bytes
Created by:	On the RACF database by ICHMIN00, IRRMIN00, ICHUT400, IRRUT400. In storage by RACF initialization if the database is not shared.
Pointed to by:	DSDEHDR in DSST data area
Serialization:	(None)
Function:	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.

Offsets					
Dec	**Hex**	**Type**	**Len**	**Name (Dim)**	**Description**
0	(0)	STRUCTURE	4096	INVICB	INVENTORY CONTROL BLOCK
0	(0)	CHARACTER	1048	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
		1...		ICBEXTND	RACF DATA SET HAS BEEN EXTENDED RACF WILL NO LONGER PROCESS UPDATES TO THIS VERSION
		.1..		ICBUPDAT	VSAMDSET GROUP MUST BE UPDATED
		..1.		ICBEDAT	RESIDENT DATA BLOCKS OPTION USED ON SOME SYSTEM
		...1		ICBDSFMT	1 = new racf ds format
		 1111		*	RESERVED
27	(1B)	ADDRESS	1	ICTMPCNT	NUMBER OF TEMPLATES
28	(1C)	BITSTRING	6	ICBAMHWM	BAM HWM.
34	(22)	CHARACTER	16	ICBTEMP (10)	SPACE FOR 10 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
194	(C2)	CHARACTER	1	ICBSTAT	STATUS

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
		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
		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 DATASET CLASS
		...1		ICBDGCM	GENERIC COMMANDS FOR DATASET 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
		.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

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
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.
337	(151)	BITSTRING	4	ICBCGEN	GENERIC PROFILE CHECKING, CLASS MASK
341	(155)	BITSTRING	4	ICBCGCM	GENERIC COMMAND PROCESSING, CLASS MASK
345	(159)	BITSTRING	4	ICBMOPT	MISC OPTIONS.
		1...		ICBFPDS	FASTPATH FOR DATASET CLS.
		.1..		ICBTDSN	TAPE DATA SET PROTECTION IN EFFECT
		..11 1111		*	RESERVED
		1...		ICBPRO	PROTECT-ALL IS IN EFFECT
		.1..		ICBPROF	1 - PROTECT-ALL WARNING IN EFFECT 0 - PROTECT-ALL FAILURE 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)
		.... .111		*	RESERVED
		1...		ICBPROG	PROGRAM CONTROL IS IN EFFECT
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 DATASET NAMES
357	(165)	CHARACTER	9	ICBQUAL	INSTALLATION CONTROLLED PREFIX FOR SINGLE LEVEL DATASET NAMES, PLUS PERIOD FOR LEVEL
366	(16E)	UNSIGNED	1	ICBSLAU	SECLEVEL TO AUDIT
367	(16F)	CHARACTER	1	*	RESERVED

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
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
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
517	(204)	ADDRESS	1	ICNTMPCT	NUMBER OF NEW TEMPLATES
518	(206)	CHARACTER	16	ICBNTMPVR (10)	Template version levels
518	(206)	CHARACTER	8	ICTMPRL	Release level
526	(20E)	CHARACTER	8	ICTMPAL	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 WHICH CONTAINS THE PROTECT 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

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
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(user-ID))
868	(364)	CHARACTER	8	ICBJUND	USER-ID from the SETROPTS command JES(UNDEFINEDUSER(user-ID))
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	SETR MLACTIVE FAILURES/WARNING - FAILURES if ″ON″ - (1) - WARNING if ″OFF″ - (0)
		...1		ICBCATF	SETROP CATDSNS FAILURES/ WARNING - FAILURES if ″ON″ - (1) - WARNING if ″OFF″ - (0)
		.... 1...		ICBAAPL	SETROPTS APPLAUDIT ON if active
		.... .111		*	Reserved
882	(372)	CHARACTER	8	ICBSPAUD	VM/SP PROFILE NAME WHICH CONTAINS THE AUDIT SETTINGS
890	(37A)	CHARACTER	8	ICBSPCTL	VM/SP PROFILE NAME 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
		...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)	CHARACTER	104	*	Reserved for future use

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
1010	(3F2)	CHARACTER	8	ICBID	Identifier, gets filled in during IRRMIN00 initialization
1018	(3FA)	CHARACTER	6	ICBVRMN	Version/release/modification #, also filled in IRRMIN00
					End of 1st 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
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 of 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
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)	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 1.9 for ICBVRMN field
6	CHARACTER	1092	ICB1092	Constant representing 1.9.2 for ICBVRMN field
6	CHARACTER	1100	ICB1100	Constant representing 1.10 for ICBVRMN field (VM)
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
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

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
ICBAMRBA	14		3
ICBAOPR	C5	01	4

Name	Hex Offset	Hex Value	Level
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
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

Name	Hex Offset	Hex Value	Level
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
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
ICBNTVS	C2	10	4
ICBPELEN	101		4
ICBPHIST	FD		3
ICBPINV	F0		3
ICBPRMSG	2C1		4
ICBPRO	15A	80	4

Name	Hex Offset	Hex Value	Level
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
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

Name	Hex Offset	Hex Value	Level
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
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

Notices

This information was developed for products and services offered in the U.S.A. IBM may not offer the products, services, or features discussed in this document in all 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
Armonk, New York 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106-0032, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:
INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs

and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
Mail Station P300
2455 South Road
Poughkeepsie, New York 12601-5400
U.S.A.
Attention: Information Request

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 measurement 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 may contain examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information may contain sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

Programming Interface Information

This publication, *z/VM: RACF Security Server Diagnosis Guide* documents information NOT intended to be used as Programming Interfaces of RACF Security Server for z/VM.

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” at www.ibm.com/legal/copytrade.shtml

Adobe is a registered trademark of Adobe Systems Incorporated in the United States, and/or other countries.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Other company, product, and service names may be trademarks or service marks of others.

Glossary

For a list of z/VM terms and their definitions, see *z/VM: Glossary*.

The z/VM glossary is also available through the online z/VM HELP Facility. For example, to display the definition of the term “dedicated device”, issue the following HELP command:

```
help glossary dedicated device
```

While you are in the glossary help file, you can do additional searches:

- To display the definition of a new term, type a new HELP command on the command line:

  ```
  help glossary newterm
  ```

  This command opens a new help file inside the previous help file. You can repeat this process many times. The status area in the lower right corner of the screen shows how many help files you have open. To close the current file, press the Quit key (PF3/F3). To exit from the HELP Facility, press the Return key (PF4/F4).
- To search for a word, phrase, or character string, type it on the command line and press the Clocate key (PF5/F5). To find other occurrences, press the key multiple times.

  The Clocate function searches from the current location to the end of the file. It does not wrap. To search the whole file, press the Top key (PF2/F2) to go to the top of the file before using Clocate.

Bibliography

See the following publications for additional information about z/VM.This bibliography lists the publications in the z/VM product library plus some related publications. For abstracts of the z/VM publications, see *z/VM: General Information*.

Where to Get z/VM Information

z/VM product information is available from the following sources:

- z/VM Information Center at publib.boulder.ibm.com/infocenter/zvm/v6r1/index.jsp
- z/VM Internet Library at www.ibm.com/eserver/zseries/zvm/library/
- IBM Publications Center at www.elink.ibmlink.ibm.com/publications/servlet/pbi.wss
- *IBM Online Library: z/VM Collection on DVD*, SK5T-7054

z/VM Base Library

Overview

- *z/VM: General Information*, GC24-6193
- *z/VM: Glossary*, GC24-6195
- *z/VM: License Information*, GC24-6200

Installation, Migration, and Service

- *z/VM: Guide for Automated Installation and Service*, GC24-6197
- *z/VM: Migration Guide*, GC24-6201
- *z/VM: Service Guide*, GC24-6232
- *z/VM: VMSES/E Introduction and Reference*, GC24-6243

Planning and Administration

- *z/VM: CMS File Pool Planning, Administration, and Operation*, SC24-6167
- *z/VM: CMS Planning and Administration*, SC24-6171
- *z/VM: Connectivity*, SC24-6174
- *z/VM: CP Planning and Administration*, SC24-6178
- *z/VM: Getting Started with Linux on System z*, SC24-6194
- *z/VM: Group Control System*, SC24-6196
- *z/VM: I/O Configuration*, SC24-6198
- *z/VM: Running Guest Operating Systems*, SC24-6228
- *z/VM: Saved Segments Planning and Administration*, SC24-6229
- *z/VM: Secure Configuration Guide*, SC24-6230
- *z/VM: TCP/IP LDAP Administration Guide*, SC24-6236
- *z/VM: TCP/IP Planning and Customization*, SC24-6238
- *z/OS and z/VM: Hardware Configuration Manager User's Guide*, SC33-7989

Customization and Tuning

- *z/VM: CP Exit Customization*, SC24-6176
- *z/VM: Performance*, SC24-6208

Operation and Use

- *z/VM: CMS Commands and Utilities Reference*, SC24-6166
- *z/VM: CMS Pipelines Reference*, SC24-6169
- *z/VM: CMS Pipelines User's Guide*, SC24-6170
- *z/VM: CMS Primer*, SC24-6172
- *z/VM: CMS User's Guide*, SC24-6173
- *z/VM: CP Commands and Utilities Reference*, SC24-6175
- *z/VM: System Operation*, SC24-6233
- *z/VM: TCP/IP User's Guide*, SC24-6240
- *z/VM: Virtual Machine Operation*, SC24-6241
- *z/VM: XEDIT Commands and Macros Reference*, SC24-6244
- *z/VM: XEDIT User's Guide*, SC24-6245
- *CMS/TSO Pipelines: Author's Edition*, SL26-0018

Application Programming

- *z/VM: CMS Application Development Guide*, SC24-6162
- *z/VM: CMS Application Development Guide for Assembler*, SC24-6163
- *z/VM: CMS Application Multitasking*, SC24-6164
- *z/VM: CMS Callable Services Reference*, SC24-6165

- *z/VM: CMS Macros and Functions Reference*, SC24-6168
- *z/VM: CP Programming Services*, SC24-6179
- *z/VM: CPI Communications User's Guide*, SC24-6180
- *z/VM: Enterprise Systems Architecture/ Extended Configuration Principles of Operation*, SC24-6192
- *z/VM: Language Environment User's Guide*, SC24-6199
- *z/VM: OpenExtensions Advanced Application Programming Tools*, SC24-6202
- *z/VM: OpenExtensions Callable Services Reference*, SC24-6203
- *z/VM: OpenExtensions Commands Reference*, SC24-6204
- *z/VM: OpenExtensions POSIX Conformance Document*, GC24-6205
- *z/VM: OpenExtensions User's Guide*, SC24-6206
- *z/VM: Program Management Binder for CMS*, SC24-6211
- *z/VM: Reusable Server Kernel Programmer's Guide and Reference*, SC24-6220
- *z/VM: REXX/VM Reference*, SC24-6221
- *z/VM: REXX/VM User's Guide*, SC24-6222
- *z/VM: Systems Management Application Programming*, SC24-6234
- *z/VM: TCP/IP Programmer's Reference*, SC24-6239
- *Common Programming Interface Communications Reference*, SC26-4399
- *Common Programming Interface Resource Recovery Reference*, SC31-6821
- *z/OS: IBM Tivoli Directory Server Plug-in Reference for z/OS*, SA76-0148
- *z/OS: Language Environment Concepts Guide*, SA22-7567
- *z/OS: Language Environment Debugging Guide*, GA22-7560
- *z/OS: Language Environment Programming Guide*, SA22-7561
- *z/OS: Language Environment Programming Reference*, SA22-7562
- *z/OS: Language Environment Run-Time Messages*, SA22-7566
- *z/OS: Language Environment Writing ILC Applications*, SA22-7563
- *z/OS MVS Program Management: Advanced Facilities*, SA22-7644
- *z/OS MVS Program Management: User's Guide and Reference*, SA22-7643

Diagnosis

- *z/VM: CMS and REXX/VM Messages and Codes*, GC24-6161
- *z/VM: CP Messages and Codes*, GC24-6177
- *z/VM: Diagnosis Guide*, GC24-6187
- *z/VM: Dump Viewing Facility*, GC24-6191
- *z/VM: Other Components Messages and Codes*, GC24-6207
- *z/VM: TCP/IP Diagnosis Guide*, GC24-6235
- *z/VM: TCP/IP Messages and Codes*, GC24-6237
- *z/VM: VM Dump Tool*, GC24-6242
- *z/OS and z/VM: Hardware Configuration Definition Messages*, SC33-7986

z/VM Facilities and Features

Data Facility Storage Management Subsystem for VM

- *z/VM: DFSMS/VM Customization*, SC24-6181
- *z/VM: DFSMS/VM Diagnosis Guide*, GC24-6182
- *z/VM: DFSMS/VM Messages and Codes*, GC24-6183
- *z/VM: DFSMS/VM Planning Guide*, SC24-6184
- *z/VM: DFSMS/VM Removable Media Services*, SC24-6185
- *z/VM: DFSMS/VM Storage Administration*, SC24-6186

Directory Maintenance Facility for z/VM

- *z/VM: Directory Maintenance Facility Commands Reference*, SC24-6188
- *z/VM: Directory Maintenance Facility Messages*, GC24-6189
- *z/VM: Directory Maintenance Facility Tailoring and Administration Guide*, SC24-6190

Open Systems Adapter/Support Facility

- *System z10, System z9 and eServer zSeries: Open Systems Adapter-Express Customer's Guide and Reference*, SA22-7935

- *System z9 and eServer zSeries 890 and 990: Open Systems Adapter-Express Integrated Console Controller User's Guide*, SA22-7990
- *System z: Open Systems Adapter-Express Integrated Console Controller 3215 Support*, SA23-2247

Performance Toolkit for VM™

- *z/VM: Performance Toolkit Guide*, SC24-6209
- *z/VM: Performance Toolkit Reference*, SC24-6210

RACF® Security Server for z/VM

- *z/VM: RACF Security Server Auditor's Guide*, SC24-6212
- *z/VM: RACF Security Server Command Language Reference*, SC24-6213
- *z/VM: RACF Security Server Diagnosis Guide*, GC24-6214
- *z/VM: RACF Security Server General User's Guide*, SC24-6215
- *z/VM: RACF Security Server Macros and Interfaces*, SC24-6216
- *z/VM: RACF Security Server Messages and Codes*, GC24-6217
- *z/VM: RACF Security Server Security Administrator's Guide*, SC24-6218
- *z/VM: RACF Security Server System Programmer's Guide*, SC24-6219
- *z/VM: Security Server RACROUTE Macro Reference*, SC24-6231

Remote Spooling Communications Subsystem Networking for z/VM

- *z/VM: RSCS Networking Diagnosis*, GC24-6223
- *z/VM: RSCS Networking Exit Customization*, SC24-6224
- *z/VM: RSCS Networking Messages and Codes*, GC24-6225
- *z/VM: RSCS Networking Operation and Use*, SC24-6226
- *z/VM: RSCS Networking Planning and Configuration*, SC24-6227
- *Network Job Entry: Formats and Protocols*, SA22-7539

Prerequisite Products

Device Support Facilities

- *Device Support Facilities: User's Guide and Reference*, GC35-0033

Environmental Record Editing and Printing Program

- *Environmental Record Editing and Printing Program (EREP): Reference*, GC35-0152
- *Environmental Record Editing and Printing Program (EREP): User's Guide*, GC35-0151

Index

A

abend
  symptoms for a complete problem search 43
ACEE data area
  control block overview on z/VM 136
ALL keyword
  on LIST subcommand 74
analyzing
  console logs 40
ASCB data area
  control block overview on z/VM 136
ASXB data area
  control block overview on z/VM 136
  overview 136
AUDIT request
  parameter list 91
AUTH requests
  parameter list 92

B

BAM blocks, RACF database
  block masks 59
    example of 60
  description of 57, 59
  format of the header 59
BLKUPD command
  format of 71
  invoking BLKUPD utility 70
BLKUPD utility
  considerations when using 70
  description of 70
  END command 72
  invoking the 71
  LOCATE command 71
  locating an index entry 71
  READ subcommand 72
  terminating the 72
  updating a RACF block 72

C

CHANGE subcommand of DISPLAY subcommand
  format of 77
checking the parameter list passed to RACF macros
  analyzing on z/VM 42
  checking on z/VM 41
  checking parameter lists on z/VM 41
  wait problem on z/VM 42
class identifier, RACF database
  adding to profile names 66
CLASS keyword
  on DISPLAY subcommand 76, 77, 79
classname
  on LOCATE command 72
collecting problem data for RACF 39
COMP keyword
  on CHANGE subcommand of DISPLAY subcommand 77
  on INSERT subcommand of DISPLAY subcommand 78
component identification for RACF
  on z/VM 36
console logs, analyzing 40
control block overview
  on z/VM 134
creating a sequential file for further manipulation
  using the database unload utility (IRRDBU00) 57
CVT (communication vector table)
  overview of 135

D

Data Encryption Standard (DES) algorithm 28
database unload utility (IRRDBU00)
  using the 57
DEFINE request
  parameter list 96
DELETE subcommand of DISPLAY subcommand 80
DES (Data Encryption Standard) algorithm 28
diagnosis
  problems in RACF 3
  procedures for RACF 4
DIRAUTH request
  parameter list 102
DISPLAY subcommand of READ command
  format of 76
  subcommands of DISPLAY 79
documentation error
  example of search argument (z/VM) 37
dump viewing facility on VM/ESA 10
  troubleshooting table 10, 21, 22
DUPLICATE
  on CHANGE subcommand of DISPLAY subcommand 77
  on INSERT subcommand of DISPLAY subcommand 78

E

END command 72
END subcommand of DISPLAY subcommand
  format of 80
END subcommand of READ command 81
ENTRY keyword
  on CHANGE subcommand of DISPLAY subcommand 77
  on DISPLAY subcommand 76, 77, 78
  on INSERT subcommand of DISPLAY subcommand 78
entryname
  on LOCATE command 72

EXTRACT request
parameter list 103

F

failure keywords
for a complete search 43
for a preliminary search 34
FASTAUTH request
parameter list 105
FIND subcommand of READ command
format of 74
finding RACF information 3
FORMAT subcommand of READ command
format of 75
list of the information printed 75
free-format search
examples of symptoms for RACF 34, 43

G

GENERIC keyword
on CHANGE subcommand of DISPLAY subcommand 77
on DISPLAY subcommand 76, 78, 79
on INSERT subcommand of DISPLAY subcommand 78
on LOCATE command 72

H

header block (ICB on RACF database)
description of 57, 58
format 58
header, RACF database
for BAM blocks 59
for index blocks 62

I

ICB (inventory control block on RACF database)
description of 57, 58
format 58
ICHDEX01 exit routine
DMKRWA in nonpageable part of CP nucleus 28
installed correctly 28
troubleshooting table 29
ICHEINTY macro
common problems 8
parameter list 117
incorrect output
example of search argument (z/VM) 37
symptoms for a complete problem search 43
index blocks
information listed by the FORMAT subcommand 75
index blocks, RACF database
compression of index entries 65
description of 60
format of 61
header 62
index blocks, RACF database *(continued)*
format of *(continued)*
index entries 63
format of a RACF database
sequence set chain pointers 65
how profiles are located 60
index compression on RACF database
types of
upper-level index name 66
index compression, RACF database
types of 65
front-end 66
index entries
locating in a sequence set 71
index entry table, RACF database
format of
sequence set chain pointers 65
index structure on RACF database
example of 61
Information/Access
use for RACF problem searches 34
Information/Management
use for RACF problem searches 34
Information/MVS
use for RACF problem searches 34
Information/System
use for RACF problem searches 34
INSERT subcommand of DISPLAY subcommand
format of 78
inventory control block (ICB) on RACF database
description of 57
format of 57
IRRDBU00 (database unload utility)
using the 57
IRRTEMP2 member of SYS1.MACLIB, RACF database
used by IRRMIN00 on RACF database 58
IRRUT100 utility
common problems 5
IRRUT200 utility
common problems 5
IRRUT400 utility
common problems 6

L

level of failing module 43
LIST request
parameter list 106
LIST subcommand of READ command
format of 74
LISTALL keyword
on LOCATE command 72
LOCATE command
format of 72
locating an index entry 72
loop
example of search argument (z/VM) 37
hang 34
in RACF
diagnostic procedures 29
return codes 33

loop *(continued)*
symptoms for a complete problem search 43
troubleshooting table 30, 32, 33
troubleshooting table for waits on z/VM 34
wait 34

M

masks, RACF database
blocks masks in the RACF database 59
message
example of search argument (z/VM) 37

N

NEW keyword
on FIND subcommand 74
on FORMAT subcommand 75
on LIST subcommand 74
NEXT subcommand of DISPLAY subcommand 79
NODUPLICATE
on CHANGE subcommand of DISPLAY subcommand 77
on INSERT subcommand of DISPLAY subcommand 78
non-VSAM data set
used for the RACF database 57
NOSAVE keyword
descriptions 87
on END subcommand 81
on END subcommand of DISPLAY subcommand 80

O

OFFSET keyword
on REPLACE subcommand 75
OLD keyword
on FIND subcommand 74
on FORMAT subcommand 75
on LIST subcommand 74

P

parameter list
AUDIT service 91
AUTH service 92
DEFINE service 96
DIRAUTH service 102
EXTRACT service 103
FASTAUTH service 105
ICHEINTY 117
LIST service 106
RACROUTE REQUEST=AUDIT 91
RACROUTE REQUEST=AUTH 92
RACROUTE REQUEST=DEFINE 96
RACROUTE REQUEST=DIRAUTH 102
RACROUTE REQUEST=EXTRACT 103
RACROUTE REQUEST=FASTAUTH 105
RACROUTE REQUEST=LIST 106
RACROUTE REQUEST=STAT 107
parameter list *(continued)*
RACROUTE REQUEST=TOKENBLD 108
RACROUTE REQUEST=TOKENEXTR 112
RACROUTE REQUEST=TOKENMAP 112
RACROUTE REQUEST=VERIFY 113
RACROUTE REQUEST=VERIFYX 113
STAT service 107
TOKENBLD service 108
TOKENEXTR service 112
TOKENMAP service 112
VERIFY service 113
VERIFYX service 113
performing a preliminary search for a matching problem 34
preliminary search for a matching problem 34
problem determination for a RACF service machine 136
problem diagnosis
for RACF 3
making sure this is a RACF problem 4
procedures for RACF 4
problem reporting procedure for RACF 45
worksheet for reporting problems 46, 47, 48, 49, 50, 51, 52, 53, 54
process flow
CP commands 123, 124
DIAGNOSE codes 123
DIAGNOSE X'A0' 125
system functions 123
process flows for RACF
on z/VM 121
RACF macros 126
RACROUTE macro 127
RACF command session 121
profile, RACF database
class identifier 66
description of 66
description of templates 58
displayed by BLKUPD, example of 68
how RACF locates on database 60
programming interface information 159

R

RAC command processor
auditing, process flow of 123
process flow 123
process flow on z/VM 122
RACF
collecting problem data for 39
diagnosis hints 4
finding RACF information 3
making sure this is a RACF problem 4
problem determination flowchart 136
problem diagnosis
making sure this is a RACF problem 3
procedures for RACF 3
reporting problems to IBM 45
searching for a matching problem 43
service machine
problem determination for a 136

RACF *(continued)*
starting diagnosis 3
troubleshooting problems 4
utilities
BLKUPD 70
IRRDBU00 69
RACF blocks
updating 72
RACF commands
process flow on z/VM 121
RACF database
index blocks 61
index entries 63
modifying blocks with BLKUPD 70
See RACF database 57
sequence set chain pointers 65
types of records in
BAM blocks 57, 59
header block 57, 58
index blocks 57, 60
profiles 57
templates 57, 58
using READ to update a block 72
using utilities on
BLKUPD 70
RACF database profile
class identifier 66
description of 66
description of templates 58
displayed by BLKUPD, example of 68
how RACF locates on database 60
RACF database, RACF
how RACF creates and names profiles on z/OS 67
RACF MACLIB on z/VM, RACF database
IRRTEMP2 member on z/VM 58
RACF macros
common problems 7
ICHEINTY parameter list 117
process flow on z/VM 126
RACF manager, RACF database
adding a class identifier to profile names 66
RACF on z/VM
component identifier for 36
RACF utilities
common problems 5
RACHECK macro
common problems 8
RACROUTE macro
process flow on z/VM 127, 131, 132, 133
RACF command session 128, 130
RACROUTE REQUEST=AUTH macro
parameter list 92
RACROUTE REQUEST=DEFINE macro
parameter list 96
RACROUTE REQUEST=EXTRACT macro
parameter list 103
RACROUTE REQUEST=FASTAUTH macro
parameter list 105
RACROUTE REQUEST=LIST macro
parameter list 106
RACROUTE REQUEST=STAT macro 107
RACROUTE REQUEST=STAT macro *(continued)*
parameter list 107
RACROUTE REQUEST=TOKENBLD macro
parameter list 108
RACROUTE REQUEST=TOKENMAP/TOKENXTR macros
parameter list 112
RACROUTE REQUEST=VERIFY macro
parameter list 113
RANGE keyword
on LIST subcommand 74
rba
on READ command 73
RBA (relative byte address), RACF database
how used in locating a profile on RACF database 60
RBA keyword
on CHANGE subcommand of DISPLAY subcommand 77
on INSERT subcommand of DISPLAY subcommand 78
RCVT data area
control block overview on z/VM 135
READ command
DISPLAY subcommand 79
END subcommand 81
FIND subcommand 73
FORMAT subcommand 75
LIST subcommand 74
REPLACE subcommand 75
REREAD subcommand 81
READ subcommand
for reading a RACF database block 73
for updating a RACF database block 73
format of 73
reading every profile in your database
using the database unload utility (IRRDBU00) 57
records, RACF database
in the RACF database
BAM blocks 57, 59
header block 57, 58
index blocks 57, 60
profiles 57, 66
templates 57, 58
REPLACE subcommand of READ command
format of 75
reporting problems to IBM
information to collect 45
REREAD subcommand of READ command 81
return code
example of search argument (z/VM) 37
REVOKE attribute 28

S

SAF router vector table
control block overview on z/VM 135
SAVE keyword
on END subcommand 81
on END subcommand of DISPLAY subcommand 80

search arguments
examples 34
search symptoms
for a complete search 43
search symptoms for a preliminary search 34
sequence set
locating an index entry 71
sequence set on RACF database
definition of 60
format of the chain pointers 65
STAT request
parameter list 107
string
on FIND subcommand 74
on REPLACE subcommand 75
structured search
examples of symptoms for RACF 34, 43
symptom record 10
symptoms
for a complete search 43
for a preliminary search 34
troubleshooting for RACF 9
SYS1.MACLIB, RACF database
IRRTEMP2 member on z/OS 58

T

tape volumes, RACF database
how RACF creates and names profiles on z/OS 67
templates on RACF database
description of 57
templates, RACF database
description of 58
TOKENBLD request
parameter list 108
TOKENEXTR request
parameter list 112
TOKENMAP request
parameter list 112
trademarks 161
troubleshooting table
abends invalidly issued 22
access with no message issued 22
DSMON report 22
incorrect output 22
macro return code or reason code 22
messages 22
RACF database 22
RACF LIST command output 22
reason code from macro or utility 22
report writer report 22
return code from macro or utility 22
return code from RACF report writer
RACF report writer return code 22
SMF records 22
utility return code or reason code 22

U

unloading your RACF database to a sequential file
using the IRRDBU00 utility 57
UPDATE keyword
on READ subcommand 73
utilities
BLKUPD 70
IRRDBU00 69

V

VER keyword
on REPLACE subcommand 76
VERIFY request
parameter list 113
VERIFYX request
parameter list 113

W

waits
example of search argument (z/VM) 37
symptoms for a complete problem search 43

Z

z/VM initialization 136

Program Number: 5741-A07

Printed in USA

GC24-6214-00