## Contents

**Figures** .................................................. vii

**Tables** ................................................... ix

**About this document** ....................................... xiii
Purpose of this document ....................................... xiii
Who should use this document ................................... xiii
How to use this document ....................................... xiii
Where to find more information .................................. xiii
  - Softcopy documents ........................................ xiii
  - RACF courses ............................................. xiv
IBM systems center publications ................................ xiv
Other sources of information .................................... xv
  - IBM discussion areas ...................................... xv
  - Internet sources ......................................... xv
  - The z/OS Basic Skills Information Center ................... xvi
To request copies of IBM publications ............................ xvii

**Summary of changes** ...................................... xix

**Chapter 1. Introduction** .................................. 1
Summary of commands and their functions ....................... 1

**Chapter 2. Basic information for issuing RACF commands** ... 7
How to enter RACF commands ..................................... 7
  - RACF TSO commands ......................................... 7
  - RACF operator commands ................................... 7
Command direction and automatic command direction .......... 7
  - RACF parameter library .................................... 8
  - R_admin callable service .................................. 8
  - Summary of issuing options ................................ 8
Syntax of RACF commands and operands ........................ 9
Return codes from RACF commands ................................ 11
RACF command limits for non-base segments in RACF profiles . 11
Installation exit routines from RACF commands ................ 12
Attribute and authority summary ................................ 12
  - Group authorities ........................................... 13
  - Access authority for data sets ............................ 13

**Chapter 3. RACF TSO commands** ............................. 15
How to enter RACF TSO commands ................................ 16
  - Choosing between Using RACF TSO Commands and ISPF Panels 16
Entering RACF TSO commands in the foreground ................ 17
Entering RACF TSO commands in the background ................ 18

**Chapter 4. RACF operator commands** ........................ 21
Rules for entering RACF operator commands ..................... 21

**Chapter 5. RACF command syntax** ............................ 23
ADDCROUP (Add group profile) .................................. 24
ADDS (Add data set profile) .................................... 33
ADDS (Add data set profile) .................................... 48
ALTDSD (Alter data set profile) ................................ 94

© Copyright IBM Corp. 1994, 2009
<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALTGROUP</td>
<td>Alter group profile</td>
</tr>
<tr>
<td>ALTUSER</td>
<td>Alter user profile</td>
</tr>
<tr>
<td>CONNECT</td>
<td>Connect user to group</td>
</tr>
<tr>
<td>DELDSD</td>
<td>Delete data set profile</td>
</tr>
<tr>
<td>DELGROUP</td>
<td>Delete group profile</td>
</tr>
<tr>
<td>DELUSER</td>
<td>Delete user profile</td>
</tr>
<tr>
<td>DISPLAY</td>
<td>Display signed-on-from list</td>
</tr>
<tr>
<td>HELP</td>
<td>Obtain RACF help</td>
</tr>
<tr>
<td>LISTDSD</td>
<td>List data set profile</td>
</tr>
<tr>
<td>LISTGRP</td>
<td>List group profile</td>
</tr>
<tr>
<td>LISTUSER</td>
<td>List user profile</td>
</tr>
<tr>
<td>PASSWORD or PHRASE</td>
<td>Specify user password or password phrase</td>
</tr>
<tr>
<td>PERMIT</td>
<td>Maintain resource access lists</td>
</tr>
<tr>
<td>RACDCERT</td>
<td>Manage RACF digital certificates</td>
</tr>
<tr>
<td>RACDCERT ADD</td>
<td>Add certificate</td>
</tr>
<tr>
<td>RACDCERT ADDRING</td>
<td>Add key ring</td>
</tr>
<tr>
<td>RACDCERT ADDTOKEN</td>
<td>Add token</td>
</tr>
<tr>
<td>RACDCERT ALTER</td>
<td>Alter certificate</td>
</tr>
<tr>
<td>RACDCERT ALTMAP</td>
<td>Alter mapping</td>
</tr>
<tr>
<td>RACDCERT BIND</td>
<td>Bind certificate to token</td>
</tr>
<tr>
<td>RACDCERT CHECKCERT</td>
<td>Check certificate</td>
</tr>
<tr>
<td>RACDCERT CONNECT</td>
<td>Connect a certificate to key ring</td>
</tr>
<tr>
<td>RACDCERT DELETE</td>
<td>Delete certificate</td>
</tr>
<tr>
<td>RACDCERT DELMAP</td>
<td>Delete mapping</td>
</tr>
<tr>
<td>RACDCERT DELRING</td>
<td>Delete key ring</td>
</tr>
<tr>
<td>RACDCERT DELTOKEN</td>
<td>Delete token</td>
</tr>
<tr>
<td>RACDCERT EXPORT</td>
<td>Export certificate package</td>
</tr>
<tr>
<td>RACDCERT GENCERT</td>
<td>Generate certificate</td>
</tr>
<tr>
<td>RACDCERT GENREQ</td>
<td>Generate request</td>
</tr>
<tr>
<td>RACDCERT IMPORT</td>
<td>Import certificate</td>
</tr>
<tr>
<td>RACDCERT LIST</td>
<td>List certificate</td>
</tr>
<tr>
<td>RACDCERT LISTMAP</td>
<td>List mapping</td>
</tr>
<tr>
<td>RACDCERT LISTRING</td>
<td>List key ring</td>
</tr>
<tr>
<td>RACDCERT LISTTOKEN</td>
<td>List token</td>
</tr>
<tr>
<td>RACDCERT MAP</td>
<td>Create mapping</td>
</tr>
<tr>
<td>RACDCERT REKEY</td>
<td>Rekey certificate</td>
</tr>
<tr>
<td>RACDCERT REMOVE</td>
<td>Remove certificate from key ring</td>
</tr>
<tr>
<td>RACDCERT ROLLOVER</td>
<td>Rollover certificate</td>
</tr>
<tr>
<td>RACDCERT UNBIND</td>
<td>Unbind certificate from token</td>
</tr>
<tr>
<td>RACLICK</td>
<td>Administer user ID associations</td>
</tr>
<tr>
<td>RACMAP</td>
<td>Create, delete, or list a distributed identity filter</td>
</tr>
<tr>
<td>RACPRIV</td>
<td>Set write-down privileges</td>
</tr>
<tr>
<td>RALTER</td>
<td>Alter general resource profile</td>
</tr>
<tr>
<td>RDEFINE</td>
<td>Define general resource profile</td>
</tr>
<tr>
<td>RDELETE</td>
<td>Delete general resource profile</td>
</tr>
<tr>
<td>REMOVE</td>
<td>Remove user from group</td>
</tr>
<tr>
<td>RESTART</td>
<td>Restart RRSF functions</td>
</tr>
<tr>
<td>RLIST</td>
<td>List general resource profile</td>
</tr>
<tr>
<td>RVARY</td>
<td>Change status of RACF database</td>
</tr>
<tr>
<td>SEARCH</td>
<td>Search RACF database</td>
</tr>
<tr>
<td>SET</td>
<td></td>
</tr>
<tr>
<td>SETROPTS</td>
<td>Set RACF options</td>
</tr>
<tr>
<td>SIGNOFF</td>
<td>Sign off sessions</td>
</tr>
<tr>
<td>STOP</td>
<td>Shutdown RRSF</td>
</tr>
<tr>
<td>TARGET</td>
<td>Define RRSF nodes</td>
</tr>
<tr>
<td>Appendix A. Naming considerations for resource profiles</td>
<td>647</td>
</tr>
<tr>
<td>--------------------------------------------------------</td>
<td>-----</td>
</tr>
<tr>
<td>Profile definitions</td>
<td>647</td>
</tr>
<tr>
<td>Discrete profiles</td>
<td>647</td>
</tr>
<tr>
<td>Generic profiles</td>
<td>647</td>
</tr>
<tr>
<td>Fully-qualified generic profiles (DATASET class only)</td>
<td>647</td>
</tr>
<tr>
<td>Determining RACF protection</td>
<td>648</td>
</tr>
<tr>
<td>Profile names for data sets</td>
<td>648</td>
</tr>
<tr>
<td>Discrete profiles</td>
<td>649</td>
</tr>
<tr>
<td>Generic profile rules—enhanced generic naming inactive</td>
<td>649</td>
</tr>
<tr>
<td>Generic profile rules—enhanced generic naming active</td>
<td>650</td>
</tr>
<tr>
<td>Choosing between discrete and generic profiles</td>
<td>652</td>
</tr>
<tr>
<td>Profile names for general resources</td>
<td>653</td>
</tr>
<tr>
<td>Permitting profiles for GENERICOWNER classes</td>
<td>656</td>
</tr>
<tr>
<td>Commands to administer VM shared file system profiles</td>
<td>657</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Appendix B. Supplied RACF resource classes</th>
<th>659</th>
</tr>
</thead>
<tbody>
<tr>
<td>Supplied resource classes for z/OS systems</td>
<td>659</td>
</tr>
<tr>
<td>Supplied resource classes for z/VM systems</td>
<td>667</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Appendix C. Accessibility</th>
<th>669</th>
</tr>
</thead>
<tbody>
<tr>
<td>Using assistive technologies</td>
<td>669</td>
</tr>
<tr>
<td>Keyboard navigation of the user interface</td>
<td>669</td>
</tr>
<tr>
<td>z/OS information</td>
<td>669</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Notices</th>
<th>671</th>
</tr>
</thead>
<tbody>
<tr>
<td>Policy for unsupported hardware</td>
<td>673</td>
</tr>
<tr>
<td>Trademarks</td>
<td>673</td>
</tr>
</tbody>
</table>

| Index     | 675 |
Figures

1. Key to symbols in command syntax diagrams ........................................... 10
2. Sample ISPF panel for RACF ................................................................. 17
3. Output for ALTUSER Command for OMVS Segment ............................... 185
4. Example 1: Output for the DISPLAY Command ...................................... 209
5. Example 2: Output for the DISPLAY Command ...................................... 209
6. Example 3: Output for the DISPLAY Command ...................................... 209
7. Example 4: Output for the DISPLAY Command ...................................... 210
8. Example 5: Output for the DISPLAY Command ...................................... 210
9. Example 1: Output for the LISTSTDSD Command ................................. 223
10. Example 2: Output for the LISTSTDSD Command .................................. 225
11. Example 3: Output for the LISTSTDSD Command .................................. 225
12. Example 4: Output for the LISTSTDSD Command .................................. 226
13. Example 1: Output for LISTGRP RESEARCH ....................................... 233
14. Example 2: Output for LISTGRP * ....................................................... 234
15. Example 3: Output for LISTGRP DFPADMIN DFP ................................ 235
16. Example 4: Output for LISTGRP DFPADMIN DFP NORACF .................... 235
17. Example 5: Output for LISTGRP OMVSG1 OMVS NORACF .................... 235
18. Example 6: Output for LISTGRP NETGROUP ....................................... 236
19. Example 1: Output for LISTUSER ......................................................... 250
20. Example 2: Output for LISTUSER (IBMUSER CALTMAN DAF0) ............. 250
21. Example 3: Output for LISTUSER DAF0 TSO ....................................... 252
22. Example 4: Output for LISTUSER NORACF TSO ................................. 252
23. Example 5: Output for LISTUSER DAF0 DFP ....................................... 253
24. Example 6: Output for LISTUSER DAF0 NORACF DFP ......................... 253
25. Example 7: Output for LISTUSER DAF0 NORACF CICS ....................... 254
26. Example 8: Output for LISTUSER DAF0 NORACF LANGUAGE .................. 254
27. Example 9: Output for LISTUSER DAF0 NORACF OPERPARM ............... 254
28. Example 10: Output for listing OMVS user information .......................... 255
29. Example 11: Output for LISTUSER CSMITH OMVS NORACF (Using Defaults) .... 255
30. Example 12: Output for LISTUSER CSMITH NORACF DCE .................... 255
31. Example 13: Output for LISTUSER RONTOMS NORACF KERB ............... 255
32. Example 14: Output for LISTUSER MRSERVER PROXY NORACF ............. 256
33. Example 15: Output for LISTUSER KCROVE EIM NORACF ..................... 256
34. Example 16: Output for LISTUSER indicating that the user's password and password phrase are each enveloped .................................................. 256
35. Example 17: Output for listing CSDATA user information ....................... 256
36. Controlling access to RACDCERT functions ......................................... 281
37. Sample output from the RACDCERT CHECKCERT command .................. 309
38. Sample output from the RACDCERT LIST command showing an assigned PKDS label .......................................................... 351
39. Sample output from the RACDCERT LIST command showing a specified PKDS label .......................................................... 352
40. Sample output from the RACDCERT LIST command specifying the certificate by label .................................................. 352
41. Sample output from the RACDCERT LIST command listing all certificates owned by the command issuer .................................................. 353
42. Sample output from the RACDCERT LIST command for a CERTAUTH certificate .......................................................... 354
43. Sample output from the RACDCERT LIST command showing a UTF-8 or BMP character that does not map to the IBM-1047 code page .................................................. 354
44. Sample output from the LISTMAP command ......................................... 357
45. Sample output from the LISTMAP LABEL command ............................. 357
46. Sample output from the RACDCERT LISTSTRING command ................ 360
47. Sample output of RACF details from the RACDCERT LISTS_TOKEN command .......................................................... 363
48. Example 1: Output for the RACLITE LIST Command ............................. 391
49. Example 1: Output for the RACMAP LISTMAP command ....................... 398
50. Example 2: Output for the RACMAP LISTMAP command ....................... 398
<table>
<thead>
<tr>
<th></th>
<th>Example</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>51</td>
<td>Example 3: Output for the RACMAP LISTMAP command</td>
<td>398</td>
</tr>
<tr>
<td>52</td>
<td>Example 1: Output for the RLIST command</td>
<td>539</td>
</tr>
<tr>
<td>53</td>
<td>Example 2: Output for the RLIST command</td>
<td>540</td>
</tr>
<tr>
<td>54</td>
<td>Example 3: Output for the RLIST command with RESGROUP option</td>
<td>540</td>
</tr>
<tr>
<td>55</td>
<td>Example 4: Output for RLIST command with masked application key</td>
<td>541</td>
</tr>
<tr>
<td>56</td>
<td>Example 5: Output for RLIST command with encrypted application key</td>
<td>541</td>
</tr>
<tr>
<td>57</td>
<td>Example 6: Output for RLIST command for the STDATA segment</td>
<td>541</td>
</tr>
<tr>
<td>58</td>
<td>Example 7: Output for RLIST command for the KERB segment</td>
<td>541</td>
</tr>
<tr>
<td>59</td>
<td>Example 8: Output for RLIST command in the PTKTDATA class</td>
<td>542</td>
</tr>
<tr>
<td>60</td>
<td>Example 9: Output for RLIST command for the EIM segment</td>
<td>542</td>
</tr>
<tr>
<td>61</td>
<td>Example 10: Output for RLIST command for the CDTINFO segment</td>
<td>543</td>
</tr>
<tr>
<td>62</td>
<td>Example 11: Output for RLIST of the ICTX segment</td>
<td>543</td>
</tr>
<tr>
<td>63</td>
<td>Example 12: Output for RLIST of the AUTHUSER segment</td>
<td>544</td>
</tr>
<tr>
<td>64</td>
<td>Example 13: Output for RLIST of the CFDEF segment</td>
<td>545</td>
</tr>
<tr>
<td>65</td>
<td>Example 14: Output for RLIST of the SIGVER segment</td>
<td>545</td>
</tr>
<tr>
<td>66</td>
<td>Example 15: Output for RLIST of the ICSF segment</td>
<td>545</td>
</tr>
<tr>
<td>67</td>
<td>Example 1: Output for the RVARY LIST command</td>
<td>553</td>
</tr>
<tr>
<td>68</td>
<td>Example 2: Output following deactivation and deallocation of RACF.PRIM1</td>
<td>553</td>
</tr>
<tr>
<td>69</td>
<td>Example 3: Output following the activation of RACF.BACK1</td>
<td>553</td>
</tr>
<tr>
<td>70</td>
<td>Example 4: Output following the RVARY SWITCH,DATASET(RACF.PRIM1) command</td>
<td>553</td>
</tr>
<tr>
<td>71</td>
<td>Example 5: Output following the RVARY NODATA SHARE command</td>
<td>554</td>
</tr>
<tr>
<td>72</td>
<td>Example 6: Output following the RVARY DATA SHARE command</td>
<td>554</td>
</tr>
<tr>
<td>73</td>
<td>Output for SET LIST Command</td>
<td>581</td>
</tr>
<tr>
<td>74</td>
<td>Output for SETROPTS LIST</td>
<td>626</td>
</tr>
<tr>
<td>75</td>
<td>Output for TARGET Command</td>
<td>644</td>
</tr>
</tbody>
</table>
## Tables

1. Functions of RACF commands ........................................... 2
2. How the RACF commands can be issued .............................. 8
3. ADDGROUP examples ..................................................... 31
4. ADDSD examples ................................................................ 45
5. ADDUSER examples ......................................................... 89
6. ALTDSD examples ............................................................ 106
7. ALTGROUP examples ......................................................... 118
8. ALTUSER examples ........................................................... 181
9. CONNECT examples .......................................................... 193
10. DELDSD examples ............................................................ 198
11. DELGROUP example .......................................................... 201
12. DELUSER examples ........................................................... 205
13. DISPLAY examples ............................................................ 208
14. HELP examples ............................................................... 212
15. LISTDSD examples ........................................................... 222
16. LISTGRP examples ........................................................... 231
17. LISTUSER examples .......................................................... 246
18. PASSWORD examples ......................................................... 261
19. PERMIT examples ............................................................ 273
20. ICSF services (resources in the CSFSERV class) used during RACDCERT command processing 279
21. Suggested generic profiles (resources in the CSFKEYS class) that authorize access to ICSF keys based on RACDCERT function and key attributes 280
22. Suggested generic profiles (resources in the CSFKEYS class) that authorize access to ICSF keys based on CONNECT attributes 280
23. Authority required for the RACDCERT ADD function .......... 287
24. RACDCERT ADD examples ............................................... 292
25. Authority required for the RACDCERT ADDRING function ... 293
26. RACDCERT ADDRING examples ........................................ 294
27. RACDCERT ADDTOKEN examples ...................................... 296
28. Authority required for the RACDCERT ALTER function .......... 297
29. RACDCERT ALTER examples ............................................. 299
30. Authority required for the RACDCERT ALTMAP function ....... 300
31. RACDCERT ALTMAP examples ........................................... 302
32. Authority required for the RACDCERT BIND function .......... 304
33. RACDCERT BIND examples ................................................ 306
34. Authority required for the RACDCERT CHECKCERT function ... 307
35. RACDCERT CHECKCERT examples ..................................... 309
36. Authority required for the RACDCERT CONNECT function—Connecting to your own key ring 311
37. Authority required for the RACDCERT CONNECT function—Connecting to another user’s key ring ................................................................. 311
38. RACDCERT CONNECT examples ........................................... 313
39. Authority required for the RACDCERT DELETE function ....... 315
40. RACDCERT DELETE examples ............................................. 316
41. Authority required for the RACDCERT DELMAP function ....... 317
42. RACDCERT DELMAP examples ........................................... 319
43. Authority required for the RACDCERT DELRING function .... 320
44. RACDCERT DELRING examples ........................................... 321
45. RACDCERT DELTOKEN examples ........................................ 323
46. Authority required for the RACDCERT EXPORT function ...... 324
47. RACDCERT EXPORT examples ............................................ 327
48. Logic for the subjectKeyIdentifier extension for GENCERT ··· 328
49. Logic for the authorityKeyIdentifier extension for GENCERT ··· 328
50. Logic for the keyUsage extension for GENCERT ····················· 329
103. After deactivating EGN—Asterisk and percent sign in the middle........................... 652
104. After deactivating EGN—Asterisk and double asterisk at the end............................ 652
105. Generic naming for general resources................................................................. 653
106. Generic naming for general resources—Percent sign, asterisk, or double asterisk at the beginning................................................................. 655
107. Generic naming for general resources—Asterisk or double asterisk at the end.................. 655
108. Generic naming for general resources—Asterisk, double asterisk, or percent sign in the middle 655
109. Permitting profile names containing asterisks (*).................................................. 656
110. Permitting profile names containing percent signs (%). ......................................... 657
111. Commands to administer shared file system (SFS) profiles...................................... 657
112. Resource classes for z/OS systems........................................................................ 659
113. Resource classes for z/VM systems........................................................................ 667
About this document

This document supports z/OS (5694-A01) and contains information about Resource Access Control Facility (RACF), which is part of z/OS Security Server.

Purpose of this document

This publication describes the syntax and the functions of the commands for RACF®. The commands are presented in alphabetical order, and the operands within each command description are presented alphabetically. Exceptions occur where operands are positional, where alternative operands are grouped together or wherever alternative operand grouping is more practical for easier understanding.

The appendixes of this document contain information on generic and discrete profiles for data sets and general resources, as well as a list of the RACF classes.

Who should use this document

This document is intended for RACF-defined users who are responsible for creating, updating, or maintaining the profiles for users, groups, data sets, and general resources in the RACF database.

Readers must be familiar with the RACF concepts and terminology. Many RACF functions also require you to understand the more detailed descriptions in z/OS Security Server RACF Security Administrator's Guide.

How to use this document

- If you want a concise list of all the RACF commands, see Chapter 1, “Introduction,” on page 1.
- If you need a general discussion on entering RACF commands, see Chapter 2, “Basic information for issuing RACF commands,” on page 7.
- If you need information on how to read syntax diagrams, see Syntax of RACF commands and operands” on page 9.
- If you want information about entering a RACF command as a RACF TSO command, see Chapter 3, “RACF TSO commands,” on page 15.
- If you want information about entering a RACF command as a RACF operator command, see Chapter 4, “RACF operator commands,” on page 21.
- If you know the command you wish to enter, but are unsure of the syntax, see the chapter that documents the appropriate command.

Where to find more information

Where necessary, this document references information in other documents. For complete titles and order numbers for all elements of z/OS®, see z/OS Information Roadmap.

Softcopy documents

The RACF library is available on the following CD-ROM, DVD, and online library collections, in both BookManager® and Portable Document Format (PDF) files. The collections include Softcopy Reader, which is a program that enables you to view the BookManager files. You can view or print the PDF files with an Adobe® Reader.
Preface

SK3T-4269  z/OS Version 1 Release 11 Collection
This collection contains the documents for z/OS Version 1 Release 11, on CD-ROM discs.

SK3T-4271  z/OS Version 1 Release 11 and Software Products DVD Collection
This collection contains the documents for z/OS Version 1 Release 11 and the libraries for multiple releases of more than 400 z/OS-related software products, on two DVDs.

SK3T-4272  z/OS Security Server RACF Collection
This softcopy collection kit contains the Security Server library for z/OS for multiple releases in both BookManager and Portable Document Format (PDF) formats. It also contains z/OS software product documents that contain substantial RACF information. This collection does not contain licensed documents.

SK3T-7876  IBM eServer™ zSeries Redbooks Collection
This softcopy collection contains a set of documents called IBM® Redbooks® that pertain to zSeries® subject areas ranging from e-business application development and enablement to hardware, networking, Linux®, solutions, security, Parallel Sysplex® and many others.

SK2T-2177  IBM Redbooks S/390 Collection
This softcopy collection contains a set of documents called IBM Redbooks that pertain to S/390® subject areas ranging from application development and enablement to hardware, networking, security, Parallel Sysplex and many others.

RACF courses

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

H3917    Basics of z/OS RACF Administration
H3927    Effective RACF Administration
ES885    Exploiting the Advanced Features of RACF
ES840    Implementing RACF Security for CICS

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

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

IBM systems center publications

IBM systems centers produce documents known as IBM Redbooks® that can help you set up and use RACF. These documents have not been subjected to any formal review nor have they been checked for technical accuracy, but they represent current product understanding (at the time of their publication) and provide valuable information on a wide range of RACF topics. They are not shipped with RACF; you must order them separately. A selected list of these documents follows. Other documents are available, but they are not included in this list, either because the information they present has been incorporated into IBM product manuals or because their technical content is outdated.
Other sources of information

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

IBM discussion areas

IBM provides ibm.servers.mvs.racf newsgroup for discussion of RACF-related topics. You can find this newsgroup on news (NNTP) server news.software.ibm.com using your favorite news reader client.

Internet sources

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

- **Online library**
  To view and print online versions of the z/OS publications, use this address:

- **Redbooks**
  The documents known as IBM Redbooks that are produced by the International Technical Support Organization (ITSO) are available at the following address:

- **Enterprise systems security**
  For more information about security on the S/390 platform, OS/390®, and z/OS, including the elements that comprise the Security Server, use this address:

- **RACF home page**
  You can visit the RACF home page on the World Wide Web using this address:

- **RACF-L discussion list**
  Customers and IBM participants may also discuss RACF on the RACF-L discussion list. RACF-L is not operated or sponsored by IBM; it is run by the University of Georgia.
  To subscribe to the RACF-L discussion and receive postings, send a note to:
  listserv@listserv.uga.edu
  Include the following line in the body of the note, substituting your first name and last name as indicated:
  subscribe racf-l first_name last_name
  To post a question or response to RACF-L, send a note, including an appropriate Subject: line, to:
  racf-l@listserv.uga.edu
Preface

- Sample code
  You can get sample code, internally-developed tools, and exits to help you use RACF. This code works in our environment, at the time we make it available, but is not officially supported. Each tool or sample has a README file that describes the tool or sample and any restrictions on its use.
  To access this code from a Web browser, go to the RACF home page and select the “Downloads” topic from the navigation bar, or go to www.ibm.com/servers/eserver/zseries/zos/racf/goodies.html.
  The code is also available from ftp.software.ibm.com through anonymous FTP.
  To get access:
  1. Log in as user anonymous.
  2. Change the directory, as follows, to find the subdirectories that contain the sample code or tool you want to download:
     cd eserver/zseries/zos/racf/
  An announcement will be posted on the RACF-L discussion list and on newsgroup ibm.servers.mvs.racf whenever something is added.

  Note: Some Web browsers and some FTP clients (especially those using a graphical interface) might have problems using ftp.software.ibm.com because of inconsistencies in the way they implement the FTP protocols. If you have problems, you can try the following:
  - Try to get access by using a Web browser and the links from the RACF home page.
  - Use a different FTP client. If necessary, use a client that is based on command line interfaces instead of graphical interfaces.
  - If your FTP client has configuration parameters for the type of remote system, configure it as UNIX® instead of MVS™.

Restrictions
Because the sample code and tools are not officially supported,
- There are no guaranteed enhancements.
- No APARs can be accepted.

The z/OS Basic Skills Information Center
The z/OS Basic Skills Information Center is a Web-based information resource intended to help users learn the basic concepts of z/OS, the operating system that runs most of the IBM mainframe computers in use today. The Information Center is designed to introduce a new generation of Information Technology professionals to basic concepts and help them prepare for a career as a z/OS professional, such as a z/OS system programmer.

Specifically, the z/OS Basic Skills Information Center is intended to achieve the following objectives:
- Provide basic education and information about z/OS without charge
- Shorten the time it takes for people to become productive on the mainframe
- Make it easier for new people to learn z/OS.

To access the z/OS Basic Skills Information Center, open your Web browser to the following Web site, which is available to all users (no login required):
http://publib.boulder.ibm.com/infocenter/zos/basics/index.jsp
To request copies of IBM publications

Direct your request for copies of any IBM publication to your IBM representative or to the IBM branch office serving your locality.

There is also a toll-free customer support number (1-800-879-2755) available Monday through Friday from 8:30 a.m. through 5:00 p.m. Eastern Time. You can use this number to:

- Order or inquire about IBM publications
- Resolve any software manufacturing or delivery concerns
- Activate the program reorder form to provide faster and more convenient ordering of software updates
Summary of changes

Summary of changes
for SA22-7687-13 and SA22-7687-14
z/OS Version 1 Release 11

This document contains information previously presented in z/OS Security Server RACF Command Language Reference, SA22-7687-12, which supports Version 1 Release 10.

New information

• Information about the new RACMAP command is added to support distributed identity filters. The functions of the RACMAP command are:
  – MAP
  – DELMAP
  – LISTMAP
• The SIGVER operand is added and information is updated for the following commands to support digital signature verification of programs.
  – RALTER
  – RDEFINE
  – RLIST
• To enhance the management of public keys in support of ICSF, the FROMICSF operand is added for the RACDCERT GENCERT command, and the ICSF operand is added for the following commands:
  – RALTER
  – RDEFINE
  – RLIST

Changed information

• Information about the following commands is updated to include restrictions related to the new IDIDMAP class in support of distributed identity filters.
  – DELUSER
  – RALTER
  – RDEFINE
  – RDELETE
  – RLIST
  – SEARCH
  – SETROPTS
• Information about the following commands is updated to describe the new BPX.UNIQUE.USER profile in support of automatic assignment of unique UIDs and GIDs for callable services.
  – RALTER
  – RDEFINE
• "UTF-8 and BMP character restrictions" on page 278 and the descriptions of the several functions of the RACDCERT command are updated to support UTF-8 and BMP characters in digital certificate names and labels.
• The purpose of the RACDCERT command is updated in "RACDCERT (Manage RACF digital certificates)" on page 278 to document that RACF signs its certificates using a signature algorithm based on the SHA-1 standard.
• Information about the following commands is updated to support suppression of logon statistics for applications.
  – RALTER
  – RDEFINE
- SETROPTS
  - Information about the CALLABLE operand of the SET command is updated to include the new callable service IRRSPS00 in support of digital signature verification of program modules.
  - Appendix B, “Supplied RACF resource classes,” on page 659 is updated with information about the following new classes:
    - IDIDMAP
    - GXCSFKEY
    - XCSFKEY
  - Support for the following APARs is added:
    - OA26468
    - OA26536

You may notice changes in the style and structure of some content in this document—for example, headings that use upper case for the first letter of initial words only, and procedures that have a different look and format. The changes are ongoing improvements to the consistency and retrievability of information in our documents.

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

Summary of changes
for SA22-7687-12
z/OS Version 1 Release 10


New information
- The CSDATA operand is added to the following commands to support custom fields:
  - ADDGROUP
  - ADDUSER
  - ALTGROUP
  - ALTUSER
  - LISTGRP
  - LISTUSER
- The CFDEF operand is added to the following commands to support custom fields:
  - RALTER
  - RDEFINE
  - RLIST

Changed information
- "RACF command limits for non-base segments in RACF profiles" on page 11 is updated for custom fields.
- Description and output example for the KERB operand of the LISTUSER command are updated.
- The purpose and output of the LISTUSER command are updated for password phrase envelopes.
• Details about authorizations for password resetting and listing user information are updated for the following commands:
  – ALTUSER
  – LISTUSER
  – SETROPTS (MINCHANGE suboperand of the PASSWORD option)

• The description of the PCICC and ICSF operands of the RACDCERT ADD and IMPORT commands are updated to support:
  – Migrating certificates with ICSF private keys from one system to another
  – 4096-bit RSA keys

• The descriptions of the PCICC and ICSF operands of the RACDCERT GENCERT and REKEY commands are updated to support 4096-bit RSA keys.

• The description of the ALTNAME operand of the RACDCERT GENCERT command is updated to support IPv6 addresses.

• Description of the PASSWORD option for the KERB segment is updated for the following commands:
  – RALTER
  – RDEFINE

• Descriptions of the WARNING and NOWARNING options of the SETROPTS command are updated to support password phrases.

• Appendix B, “Supplied RACF resource classes,” on page 659 is retitled. This topic was previously titled “Supplied class descriptor table entries”. In addition, this topic is updated with information about new classes.

Restructured information

• To improve retrievability and clarity, most of the information previously presented in the topic entitled “RACDCERT (RACF Digital Certificate)” is now presented in a full set of topics that individually describe each RACDCERT function.

  Each topic is entitled with the name of the RACDCERT command function, and contains the purpose, authorization required, syntax, parameter descriptions, examples, and other specific information about using each function. See “RACDCERT (Manage RACF digital certificates)” on page 276.

This document contains terminology, maintenance, and editorial changes.

Summary of changes
for SA22-7687-11
z/OS Version 1 Release 9

This document contains information previously presented in z/OS Security Server RACF Command Language Reference, SA22-7687-10, which supports Version 1 Release 8.

New information

• The AES128 and AES256 options are added to the KERB ENCRYPT operands of the following commands:
  – ADDUSER
  – ALTUSER
  – RDEFINE
  – RALTER

• The following functions are added to the RACDCERT command to support z/OS PKCS #11 tokens:
  – ADDTOKEN
  – DELTOKEN
Information about new support for UTF-8 characters is added to the description of the RACDCERT command.

**Changed information**

- Descriptions for the PHRASE operand are updated for following commands:
  - ADDUSER
  - ALTUSER
  - PASSWORD or PHRASE

- Descriptions and output examples for the KERB operand are updated for the following commands:
  - LISTUSER
  - RLST

- The RACDCERT command is updated with authorization information for the new RACDCERT functions.

- The description of the KERBLVL operand of the SETROPTS command is changed. Beginning with z/OS Version 1 Release 9, the KERBLVL operand of the SETROPTS command is ignored.

- Appendix B, “Supplied RACF resource classes,” on page 659 is updated with information about new classes, including support for APAR OA20162.

- The descriptions of the NAME operand of the ADDUSER and ALTUSER commands are updated for APAR OA18327.

- The descriptions of the GENERIC and NOGENERIC operands of the SETROPTS command are updated for APAR OA19353.

This document contains terminology, maintenance, and editorial changes.
Chapter 1. Introduction

The profiles in the RACF database contain the information RACF needs to control access to resources. The RACF commands allow you to add, change, delete, and list the profiles for:

- Users
- Groups
- Data sets
- General resources, which include terminals, DASD volumes, and all other resource classes defined in the RACF class descriptor table (CDT).

Table 1 shows, in alphabetic order, each command, and its function.

Most RACF functions do not require special versions or releases of the operating system or operating system components. However, some do require that your system be at a certain level. If you are unsure about whether a particular RACF function is available with your system, see your security administrator.

Some commands require that the RACF subsystem be active or that you have authorization to issue the commands. Refer to the “Authorization Required” section with each command for details on the authorization required.

The following RACF commands are available only on RACF for VM:

- ADDFILE
- ADDDIR
- ALTFILE
- ALTDIR
- DELFILE
- DELDIR
- LFILE
- LDIRECT
- PERMFILE
- PERMDIR
- SRFFILE
- SRDIFR

See the appropriate RACF Command Language Reference for your VM system for more information.

Summary of commands and their functions

RACF commands allow you to list, modify, add, and delete profiles for users, groups, connect entries, and resources. Table 1 on page 2 shows, in alphabetic order, each of the commands and its functions.
### Table 1. Functions of RACF commands

<table>
<thead>
<tr>
<th>RACF command</th>
<th>Command functions</th>
</tr>
</thead>
</table>
| ADDGROUP     | • Define one or more new groups as a subgroup of an existing group.  
• Specify a model data set profile for a group.  
• Add a custom field for a group.  
• Define default DFP information for a group.  
• Define the z/OS UNIX information for a group.  
• Define a group as a universal group. |
| ADDSD        | • RACF-protect one or more existing data sets.  
• RACF-define one or more data sets brought from another system where they were RACF-protected.  
• RACF-define generic data set profiles.  
• Create a new data set model profile. |
| ADDUSER      | • Define one or more new users and connect the users to their default connect group.  
• Define a password, or a password and password phrase, for one or more users.  
• Specify a model data set profile for a user.  
• Define CICS® operator information.  
• Add a custom field for a user.  
• Define default DFP information for a user.  
• Define the EIM information for a user.  
• Define the OMVS information for a user.  
• Define the preferred national language.  
• Define default operator information.  
• Define default TSO logon information for a user.  
• Define default work attributes.  
• Define the z/OS UNIX information for a user.  
• Define the DCE information for a user.  
• Define default NetView® operator information.  
• Define the COMMAND field of the logon panel.  
• Define the LNOTES and NDS information for a user.  
• Define access checking with the RESTRICTED and NORESTRICTED keywords.  
• Define the KERB information for a user.  
• Define the PROXY information for a user.  
• Specify the resource security level (RSL) keys assigned to a user.  
• Specify the transaction security level (TSL) keys assigned to a user. |
| ALTDSD       | • Change one or more discrete or generic data set profiles.  
• Protect a single volume of a multivolume, non-VSAM DASD data set.  
• Remove protection from a single volume of a multivolume, non-VSAM DASD data set. |
| ALTGROUP     | • Change the information in one or more group profiles (such as the superior group, owner, or model profile name).  
• Change or delete a custom field for a group.  
• Change or delete the default DFP information for a group.  
• Add, change, or delete the information for the z/OS UNIX group. |
Table 1. Functions of RACF commands (continued)

<table>
<thead>
<tr>
<th>RACF command</th>
<th>Command functions</th>
</tr>
</thead>
</table>
| ALTUSER      | - Change the information in one or more user profiles (such as the owner, universal access authority, or security level).  
               - Revoke or reestablish one or more users' privileges to access the system.  
               - Specify logging of information about the user, such as the commands the user issues.  
               - Change the password or password phrase for one or more users.  
               - Change or delete CICS operator information.  
               - Change or delete a custom field for a user.  
               - Change or delete the default DFP information for a user.  
               - Define, change or delete the EIM information for a user.  
               - Define, change or delete the OMVS information for a user.  
               - Change the preferred national language.  
               - Change or delete the default operator information.  
               - Change or delete the default TSO logon information for a user.  
               - Change or delete the default work attributes.  
               - Add, change, or delete the information for the z/OS UNIX user.  
               - Change the DCE information for a user.  
               - Change or delete NetView operator information.  
               - Manipulate the COMMAND field of the logon panel.  
               - Change the LNOTES and NDS information for a user.  
               - Change access checking methods with the RESTRICTED and NORESTRIC TED keywords.  
               - Add or change the KERB information for a user.  
               - Add or change the PROXY information for a user.  
               - Specify the complete list of resource security level (RSL) keys assigned to a user.  
               - Specify the complete list of transaction security level (TSL) keys assigned to a user. |
| CONNECT      | - Connect one or more users to a group.  
               - Modify one or more users' connection to a group.  
               - Revoke or reestablish one or more users' privileges to access the system. |
| DELDSD       | - Delete one or more discrete or generic data set profiles.  
               - Delete a discrete data set profile for a tape data set, while retaining the data set name in the TVTOC.  
               - Remove a data set profile, but leave the data set RACF-indicated, when moving a RACF-protected data set to another system that has RACF. |
| DELGROUP     | - Delete one or more groups and their relationship to the superior group. |
| DELUSER      | - Delete one or more users and remove all of their connections to RACF groups. |
| DISPLAY      | - Display users signed on to a RACF subsystem. |
| HELP         | - Display the function and proper syntax of RACF commands. |
| LISTSER      | - List the details of one or more discrete or generic data set profiles, including the users and groups authorized to access the data sets.  
               - Determine the most specific matching generic profile for a data set.  
               - Perform a local refresh of generic DATASET profiles. |
| LISTGRP      | - List the details of one or more group profiles, including the users connected to the group.  
               - List only the information contained in a specific segment (for example, OMVS or CSDATA) of the group profile.  
               - Display limited information if the group is a UNIVERSAL group. |
| LISTUSER     | - List the details of one or more user profiles, including all of the groups to which each user is connected.  
               - List only the information contained in a specific segment (for example, OMVS or CSDATA) of the user profile. |
| PASSWORD or PHRASE | - Change your own user password or password phrase.  
               - Change one or more users' change interval for passwords and password phrases.  
               - Reset one or more user passwords to their default values. |
| PERMIT       | - Give or remove authority to access a resource to specific users or groups.  
               - Change the level of access authority to a resource for specific users or groups.  
               - Copy the list of authorized users from one resource profile to another.  
               - Delete an existing access list. |
### Table 1. Functions of RACF commands (continued)

<table>
<thead>
<tr>
<th>RACF command</th>
<th>Command functions</th>
</tr>
</thead>
</table>
| **RACDCERT** | - List information about the certificates for a specified RACF-defined user ID, or your own user ID.  
- Add a certificate and associate it with a specified RACF-defined user ID, or your own user ID, and set the TRUST status.  
- Check to see if a certificate has been defined to RACF.  
- Alter the TRUST status or label for a certificate.  
- Delete a certificate.  
- List a certificate contained in a data set and determine if it is associated with a RACF-defined user ID.  
- Add or remove a certificate from a key ring.  
- Create, delete, or list a key ring.  
- Generate a public/private key pair and certificate, replicate a digital certificate with a new public/private key pair, or retire the use of an existing private key.  
- Write (export) a certificate or certificate package to a data set.  
- Create a certificate request.  
- Create, alter, delete, or list a certificate name filter (user ID mapping).  
- Add, delete, or list a z/OS PKCS #11 token.  
- Bind a certificate to a z/OS PKCS #11 token.  
- Remove (unbind) a certificate from a z/OS PKCS #11 token.  
- Import a certificate (with its private key, if present) from a z/OS PKCS #11 token and add it to RACF. |
| **RACLINK** | - Define, approve, and delete (undefine) a user ID association.  
- List information related to a user ID association.  
- Establish password synchronization between user IDs. |
| **RACMAP** | - Create an association between a distributed user identity and a RACF user ID.  
- Define, delete, and list a distributed identity filter. |
| **RACPRIV** | - List, activate, and inactivate the user’s write-down setting.  
- Reset the user’s write-down setting to the installation-defined default. |
| **RALTER** | - Change the discrete or generic profiles for one or more resources whose class is defined in the class descriptor table.  
- Maintain the global access checking tables.  
- Maintain security category and security level tables.  
- List the encryption keys used if the profile has a KERB segment.  
- Maintain CDTINFO, CFDEF, DLFDATA, SESSION, SSIGNON, and STDATA segment information in the profiles.  
- Define, change, or delete the DOMAINDN, OPTIONS, LOCALREGISTRY, KERBREGISTRY, and X509REGISTRY information in the EIM segment.  
- Change profiles associated with a SystemView® for MVS application.  
- Define, change, or delete the LDAPHOST address, BINDDN and BINDPW information in the PROXY segment.  
- Maintain configuration options for the identity cache in the ICTX segment. |
| **RDEFINE** | - RACF-protect by a discrete or generic profile any resource whose class is defined in the class descriptor table.  
- Define entries in the global access checking table.  
- Define security category and security level tables.  
- Define the encryption keys used if the profile has a KERB segment.  
- Define CDTINFO, CFDEF, DLFDATA, SESSION, SSIGNON, and STDATA segment information in the profiles.  
- Define the EIM segment and the DOMAINDN, OPTIONS, LOCALREGISTRY, KERBREGISTRY, and X509REGISTRY information for the segment.  
- Define the list of classes for which RACF is to save RACTIONed results on the RACF database.  
- Define profiles associated with a SystemView for MVS application.  
- Define the LDAPHOST address, BINDDN and BINDPW information if the profile has a PROXY segment.  
- Define configuration options for the identity cache in the ICTX segment.  
- Define the SHARED.IDS profile in the UNIXPRIV class, which controls the default behavior of adding or altering UIDs and GIDs in the OMVS segment.  
- Modify the processing of z/OS UNIX System Services by defining profiles in the UNIXPRIV class that either:  
  - Serve as system-wide options (such as the CHOWN.UNRESTRICTED or FILE.GROUPOWNER.SETGID profiles).  
  - Define an individual superuser capability (such as the ability to change file ownership using the SUPERUSER.FILESYS.CHOWN resource), which can be granted to a user instead of assigning UID(0). |
<table>
<thead>
<tr>
<th>RACF command</th>
<th>Command functions</th>
</tr>
</thead>
</table>
| **RDELETE**  |  • Remove RACF-protection from one or more resources whose class is defined in the class descriptor table.  
• Delete the global access checking tables.  
• Delete the security category and security level tables.  
• Delete a class from the list of classes for which RACF saves RACLISted results on the RACF database. |
| **REMOVE**   |  • Remove one or more users from a group and assign a new owner for any group data sets owned by the users. |
| **RESTART**  |  • Restart a function in the RACF subsystem address space.  
• Restart the connection to a specific member system on a multisystem node. |
| **RLIST**    |  • List the details of discrete or generic profiles for one or more resources whose class is defined in the class descriptor table.  
• List the contents of the CDTINFO, CFDEF, DLFDATA, SESSION, SSIGNON, and STDATA segments in the profiles.  
• List the DOMAINDN, OPTIONS, LOCALREGISTRY, KERBREGISTRY, and X509REGISTRY information if the profile has an EIM segment.  
• List configuration options for the identity cache when the profile has an ICTX segment.  
• List the encryption keys used if the profile has a KERB segment.  
• List the LDAPHOST address, BINDDN information and whether or not a BINDPW exists if the profile has a PROXY segment.  
• Perform a local refresh of generic general resource profiles. |
| **RVARY**    |  • Dynamically deactivate and reactivate the RACF function.  
• Dynamically deactivate and reactivate the RACF primary and backup database.  
• Switch the primary and backup RACF databases.  
• Deactivate resource protection, for any resource whose class is defined in the class descriptor table, while RACF is deactivated.  
• Select operational mode when RACF is enabled for sysplex communication. |
| **SEARCH**   |  • Obtain a list of RACF profile names that meet the search criteria for a class of, resources, users, or groups. These profile names can then be displayed on your terminal.  
– Profile names that contain a specific character string  
– Profiles for resources that have not been referenced for more than a specific number of days  
– Profiles that RACF recognizes as model profiles  
– Data set and general resource profiles that contain a level equal to or greater than the level you specify  
– User and resource profiles that contain a security label that matches the security label you specify.  
– User and resource profiles that contain a security level that matches the security level that you specify  
– User and resource profiles that contain an access category that matches the access category that you specify.  
– User profiles that contain an OMVS UID equal to the UID you specify.  
– Group profiles that contain an OMVS GID equal to the GID you specify.  
– Profiles for tape volumes that contain only data sets with an expiration date that matches the criteria you specify.  
– Profiles for data sets that reside on specific volumes (or VSAM data sets that are cataloged in catalogs on specific volumes).  
– Profiles for tape data sets, non-VSAM DASD data sets, or VSAM data sets.  
• Format the selected profile names with specific character strings into a series of commands or messages and retain them in a CLIST data set.  
• Create a CLIST of the RACF profile names that meet a search criteria for a class of resources. |
| **SET**      |  • List information related to RACF remote sharing facility (RRSF) on the local node.  
• List the *rrrrrrrr.aaaaaaaa* value for the template version following the FMID/APAR value.  
• Specify the name of a member of the RACF parameter library to be processed by RRSF.  
• Set tracing, including SAFTRACE, on or off for specified operands.  
• Specify options for automatic command direction. |
### Introduction

**Table 1. Functions of RACF commands (continued)**

<table>
<thead>
<tr>
<th>RACF command</th>
<th>Command functions</th>
</tr>
</thead>
</table>
| SETROPTS     | Dynamically set system-wide options relating to resource protection, specifically:  
|              | • Choose the resource classes that RACF is to protect.  
|              | • Gather and display RACF statistics.  
|              | • Set the universal access authority (UACC) for terminals.  
|              | • Set the KERBLVL to be used if a profile has a KERB segment. (Beginning with z/OS V1R9, KERBLVL setting is ignored.)  
|              | • Specify logging of certain RACF commands and events.  
|              | • Permit list-of-groups access checking.  
|              | • Display options currently in effect.  
|              | • Enable or disable generic profile checking on either a class-by-class basis.  
|              | • Control user password syntax rules.  
|              | • Activate checking for previous passwords and password phrases.  
|              | • Limit unsuccessful attempts to access the system using incorrect passwords and password phrases.  
|              | • Control maximum change interval for passwords and password phrases.  
|              | • Control minimum change interval for passwords and password phrases.  
|              | • Control mixed-case passwords.  
|              | • Warn of password expiration.  
|              | • Control global access checking for selected individual resources or generic names with selected generalized access rules.  
|              | • Set the passwords for authorizing use of the RVARY command.  
|              | • Initiate refreshing of in-storage generic profile lists and global access checking tables.  
|              | • Enable or disable shared generic profiles for general resources in common storage.  
|              | • Enable or disable shared profiles through RACLST processing for general resources.  
|              | • Activate or deactivate auditing of access attempts to RACF-protected resources based on installation-defined security levels.  
|              | • Activate enhanced generic naming.  
|              | • Control the use of automatic data set protection (ADSP).  
|              | • Activate profile modeling for GDG, group, and user data sets.  
|              | • Activate protection for data sets with single-level names.  
|              | • Control logging of real data set names.  
|              | • Control the job entry subsystem (JES) options.  
|              | • Activate tape data set protection.  
|              | • Control whether or not data sets must be RACF-protected.  
|              | • Control the erasure of scratched DASD data sets.  
|              | • Activate program control.  
|              | • Control whether a profile creator’s user ID is automatically added to the profile’s access list.  
|              | • Make the name of the local RACF registry available to EIM services.  
|              | • Control use of the dynamic class descriptor table.  
|              | • Control multilevel security options.  

| SIGNOFF      | Sign off users from a RACF subsystem.  

| STOP         | Stop the RACF subsystem address space.  

| TARGET       | List the controls and operational characteristics of the specified target RRSF nodes.  
|              | • Specify the name of the target RRSF node.  
|              | • Request an operational state for connection to the target RRSF node.  
|              | • Delete an RRSF node from the local node.  
|              | • Specify a description of the target RRSF node.  
|              | • Purge the workspace data sets managed by RRSF in the RACF subsystem address space.  
|              | • Specify the protocol type for the transport mechanism to be used in communication between the two RRSF nodes.  
|              | • Specify the relationship between the target RRSF node and the node being configured.  
|              | • Specify a prefix for the workspace data sets allocated by and used by RRSF for each target node.  
|              | • Specify the characteristics of the workspace data sets associated with the node being defined to RRSF.  
|              | • Specify the name of a multisystem node.  
|              | • Identify the main system in a multisystem RRSF node.  

**Note:** In data sharing mode or read-only mode, RACF employs global ENQs to serialize access to the RACF database before adding or removing protection from a resource. Otherwise—unless the installation has explicitly converted to GRS—RACF uses hardware RESERVE/RELEASE.
Chapter 2. Basic information for issuing RACF commands

You can use the RACF commands to add, modify, or delete RACF profiles and to define system-wide options. Before you can issue a RACF command, you must be defined to RACF with a sufficient level of authority.

How to enter RACF commands

There are several ways to enter RACF commands.

RACF TSO commands

Some RACF commands can be entered as RACF TSO commands. For information on entering RACF commands as RACF TSO commands, see Chapter 3, “RACF TSO commands,” on page 15. For a complete list of which RACF commands can be entered as RACF TSO commands, see Table 2 on page 8.

RACF operator commands

Some RACF commands can be entered as RACF operator commands. For information on entering RACF commands as RACF operator commands, see Chapter 4, “RACF operator commands,” on page 21. For a complete list of which RACF commands can be entered as RACF operator commands, see Table 2 on page 8.

Command direction and automatic command direction

With command direction, some RACF commands can be directed to run under the authority of a user ID on a remote node, or the same node. Use the AT keyword on your command for command direction. For information on command direction, see z/OS Security Server RACF Security Administrator’s Guide. For information on the AT keywords, see the eligible command descriptions. For a complete list of RACF commands that are eligible for command direction, see Table 2 on page 8.

A failure might occur while attempting to execute a command issued on one (uplevel) system and manually or automatically directed to another (downlevel) system through RACF remote sharing facility (RRSF) for any of the following reasons:

- The command references a class unknown to the target system (for example, if the class descriptor tables are different).
- The command references a segment or field unknown to the target system (for example, if the templates or dynamic parse definition are different).
- The command uses a command keyword unknown to the target (for example, if the dynamic parse definitions or command processor code is different), or if it specifies a profile or member name that is unacceptable to the target system (for example, if the class descriptor tables have different syntax requirements for profile name length or syntax).

If an out-of-synchronization condition occurs while using automatic command direction, a RACF TSO command can be directed with the ONLYAT keyword to fix the condition. The command runs on the node specified on the ONLYAT keyword and are propagated to any other node. (Note that if the AT keyword is used, the command can be propagated by automatic command direction to other nodes.)
Basic information

Information on the ONLYAT keyword, see the eligible command descriptions. For a complete list of RACF commands that are eligible for automatic command direction, see Table 2.

Some RACF TSO commands can be automatically directed to remote nodes in order to keep profiles synchronized between the nodes. For information on automatic command direction, see z/OS Security Server RACF Security Administrator’s Guide.

RACF parameter library

Some RACF commands can be processed from the RACF parameter library. For information on using the RACF parameter library, see z/OS Security Server RACF System Programmer’s Guide. For a complete list of commands that can be processed from within the RACF parameter library, see Table 2.

R_admin callable service

You can also issue commands by calling the R_admin callable service (IRRSEQ00). For more information on using this callable service, and for a complete list of commands that can be issued in this manner, see z/OS Security Server RACF Callable Services.

Summary of issuing options

Table 2 lists the ways you can enter each RACF command.

Table 2. How the RACF commands can be issued

<table>
<thead>
<tr>
<th>RACF command</th>
<th>As a RACF TSO command?</th>
<th>As a RACF operator command?</th>
<th>With command direction?</th>
<th>With automatic command direction?</th>
<th>From the RACF parameter library?</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADDGROUP</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>ADDSD</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>ADDUSER</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>ALTDSD</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>ALTGROUP</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>ALTUSER</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>BLKUPD</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>CONNECT</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>DELDSD</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>DELGROUP</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>DELUSER</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>DISPLAY</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>HELP</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>LISTDSD</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>LISTGRP</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>LISTUSER</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>PASSWORD</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>PERMIT</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>RACDCERT</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
</tbody>
</table>
### Table 2. How the RACF commands can be issued (continued)

<table>
<thead>
<tr>
<th>RACF command</th>
<th>As a RACF TSO command?</th>
<th>As a RACF operator command?</th>
<th>With command direction?</th>
<th>With automatic command direction?</th>
<th>From the RACF parameter library?</th>
</tr>
</thead>
<tbody>
<tr>
<td>RACLINK</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>RACMAP</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>RACPRIV</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>RALTER</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>RDEFINE</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>RDELETE</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>REMOVE</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>RESTART</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>RLST</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>RVARY</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>SEARCH</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>SET</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>SETROPTS</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>SIGNOFF</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>STOP</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>TARGET</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>

**Notes:**

1. Information about the block update (BLKUPD) command appears in [z/OS Security Server RACF Diagnosis Guide](https://www.ibm.com/) not this document.
2. This command cannot be directed to a remote system using the AT or ONLYAT keyword.
3. Updates made to the RACF database by this command are eligible for propagation with automatic direction of application updates based on certain RRSFDATA profiles. For details, see “Issuing options” in the command syntax topic for this command.
4. The SEARCH command is not eligible for command direction when the CLIST keyword is specified.
5. The SETROPTS LIST command without other keywords is not eligible for automatic command direction.

### Syntax of RACF commands and operands

This publication describes the syntax and function of the RACF commands. The commands are presented in alphabetical order. Each command description contains several examples.

For the key to the symbols used in the command syntax diagrams, see [Figure 1 on page 10](#).
Basic information

1. UPPERCASE LETTERS or WORDS must be coded as they appear in the syntax diagrams but do not have to be uppercase.
2. Lowercase letters or words represent variables for which you must supply a value.
3. Parentheses ( ) must be entered exactly as they appear in the syntax diagram.
4. An ellipsis ... (three consecutive periods) indicates that you can enter the preceding item more than once.
5. A single item in brackets [ ] indicates that the enclosed item is optional. Do not specify the brackets in your command.
6. Stacked items in brackets [ ] indicate that the enclosed items are optional. You can choose one or none. Do not specify the brackets in your command.
7. Stacked items in braces { } indicate that the enclosed items are alternatives. You must specify one of the items. Do not specify the braces in your command.

**Note:** When you select a bracket that contains braces, you must specify one of the alternatives enclosed within the braces.
8. Items separated by a vertical bar | indicate that you can specify only one of the items. Do not specify the vertical bar in your command.
9. An underlined operand indicates the default value when no alternate value is specified.
10. **BOLDFACE** or boldface indicates information that must be given for a command.
11. Single quotation marks ‘ ’ indicate that information must be enclosed in single quotation marks.

---

**Figure 1. Key to symbols in command syntax diagrams**

The syntax for all occurrences of the **userid**, **group-name**, **password**, **class-name**, **profile-name**, **volume-serial**, **terminal-id**, and **date** operands in this book is as follows:

**userid**

1–8 alphanumeric characters. The user ID can consist entirely of numbers and need not begin with any particular character.

For TSO users who are defined to RACF, the user ID cannot exceed seven characters and must begin with an alphabetic, # (X'7B'), $ (X'5B'), or @ (X'7C') character.

**group-name**

1–8 alphanumeric characters beginning with an alphabetic, # (X'7B'), $ (X'5B'), or @ (X'7C') character. (You can set the default prefix to a group name only if the group name contains 1–7 characters. If the group name has 8 characters, you must always enter fully-qualified group data set names on the commands.

**password**

1–8 alphanumeric characters. Each installation can define its own password syntax rules. Lowercase alphanumeric characters are valid and maintained in the case entered if SETR PASSWORD(MIXEDCASE) is in effect.

**class-name**

Valid class names are USER, GROUP, DATASET, and those classes defined in the class descriptor table.
The entries supplied by IBM in the class descriptor table are listed in Appendix B, “Supplied RACF resource classes,” on page 659.

### profile-name
Either a discrete name or a generic name, as described in Appendix A, “Naming considerations for resource profiles,” on page 647.

### terminal-id
1–8 alphanumeric characters.

### volume-serial
1–6 alphanumeric characters.

### date
RACF interprets dates as 20yy when the year is less than 71, and 19yy when the year is 71 or higher.

## Return codes from RACF commands

All of the RACF commands (except RVARY) issue the following return codes. RVARY issues return codes of 0, 8, and 12.

<table>
<thead>
<tr>
<th>Decimal code</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Normal completion.</td>
</tr>
<tr>
<td>4</td>
<td>The command encountered a user error or an authorization failure and attempted to continue processing. Refer to documentation of the error message that RACF issues to determine what part of the current entity (if any) was completed. If additional entities were specified on the command, RACF attempts to process them.</td>
</tr>
<tr>
<td>8</td>
<td>The command encountered a user error or an authorization failure and terminated processing.</td>
</tr>
<tr>
<td>12</td>
<td>The command encountered a system error and terminated processing.</td>
</tr>
</tbody>
</table>

Except for commands entered using the AT keyword or the RACLINK command, you can use CLIST processing or REXX™ exec processing to check for these return codes. Commands entered using the AT keyword or the RACLINK command run in two phases:

- The first phase validates the issuer’s authority to use the function and determines whether the RACF subsystem address space is available to handle the second phase. A return code of 0 from the first phase means the request was successfully passed to the RACF subsystem address space. A return code of 8 means the command was rejected.
- The second phase runs in the RACF subsystem address space. Return codes within the address space cannot be interrogated by the issuer’s CLIST or REXX exec. The success of the command processing within the second phase must be determined from the returned output messages, if any.

## RACF command limits for non-base segments in RACF profiles

In general, you can specify up to 255 operands with a single command to create or alter a non-base profile segment. **Exception:** For the CSDATA segment in user and group profiles, you can specify up to 85 operands with a single command.

Examples of non-base segments in user profiles include the OMVS, TSO and CSDATA segments. Examples of non-base segments in general resource profiles include the CDTINFO, EIM, and STDATA segments.
The following RACF commands create or alter non-base segments in profiles, and operate within these limits:

- **ADDGROUP**
- **ADDSD**
- **ADDUSER**
- **ALTDSD**
- **ALTGROUP**
- **ALTUSER**
- **RALTER**
- **RDEFINE**.

If you issue a single command specifying more than 255 operands for a non-base profile segment (other than the CSDATA segment), the command will not execute correctly although it will appear successful. Only the remainder of \( n \div 255 \) operands will be applied to the database, where \( n \) is the number of operands you specified for the segment. For example, if you issue the ALTUSER command specifying 259 operands for a non-base segment, RACF applies only the last 4 operands to the database.

If you issue a single command specifying more than 85 operands for the CSDATA segment, the command will not execute correctly although it will appear successful. Only the first 85 operands will be applied to the database, and the rest will be ignored. For example, if you issue the ALTUSER command specifying 93 operands for the CDTINFO segment, RACF applies only the first 85 CSDATA operands and ignores the last 8 CSDATA operands.

### Installation exit routines from RACF commands

RACF provides a general purpose exit, IRREVX01, that can be modified by installations to receive control when the following RACF TSO commands are issued.

<table>
<thead>
<tr>
<th>Command</th>
<th>Exit Routines</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADDGROUP</td>
<td>ALTUSER</td>
</tr>
<tr>
<td>ADDSD</td>
<td>LISTDS</td>
</tr>
<tr>
<td>ADDUSER</td>
<td>RALTER</td>
</tr>
<tr>
<td>ALTDSD</td>
<td>SEARCH</td>
</tr>
<tr>
<td>ALTGROUP</td>
<td>SETROPTS</td>
</tr>
<tr>
<td>ALTER</td>
<td></td>
</tr>
<tr>
<td>CONNECT</td>
<td></td>
</tr>
<tr>
<td>DELDSD</td>
<td></td>
</tr>
<tr>
<td>DELGROUP</td>
<td></td>
</tr>
<tr>
<td>DELUSER</td>
<td></td>
</tr>
<tr>
<td>LISTGRP</td>
<td></td>
</tr>
<tr>
<td>PASSWORD</td>
<td></td>
</tr>
<tr>
<td>PERMIT</td>
<td></td>
</tr>
<tr>
<td>RDELETE</td>
<td></td>
</tr>
<tr>
<td>RDEFINE</td>
<td></td>
</tr>
<tr>
<td>REMOVE</td>
<td></td>
</tr>
<tr>
<td>RLST</td>
<td></td>
</tr>
</tbody>
</table>

RACF does not invoke the IRREVX01 exit for the BLKUPD, RVARY, RACDCERT, RACMAP, RACLXNK and RACPRIV commands or for true RACF operator commands such as RESTART, TARGET, and SIGNOFF.

Your location might use installation-written exit routines to take additional security actions during RACF command processing, and these actions can affect the results you get when you issue a RACF command. For example, your location could use the ICHPWXX01 preprocessing exit to install its own routine to examine a new password and new change interval.


### Attribute and authority summary

Each command description in this book includes a section called “Authorization Required”, which describes how attributes and authorities affect your use of that command.
Group authorities

The group authorities, which define user responsibilities within the group, are shown below in order of least to most authority. Each level includes the privileges of the levels above it.

- **USE**  Allows you to access resources to which the group is authorized
- **CREATE**  Allows you to create RACF data set profiles for the group
- **CONNECT**  Allows you to connect other users to the group
- **JOIN**  Allows you to add new subgroups or users to the group, as well as assign group authorities to the new members

For more information on group authority, refer to [z/OS Security Server RACF Security Administrator’s Guide](http://z/os/secur/admin).  

Access authority for data sets

Data sets can have one of the following access authorities:

- **NONE**  Does not allow users to access the data set.
- **EXECUTE**  For a private load library, EXECUTE allows users to load and execute, but not to read or copy, programs (load modules) in the library.

  In order to specify EXECUTE for a private load library, you must ask for assistance from your RACF security administrator.

- **READ**  Allows users to access the data set for reading only. (Note that users who can read the data set can copy or print it.)

- **UPDATE**  Allows users to read from, copy from, or write to the data set. UPDATE does not authorize a user to delete, rename, move, or scratch the data set.

  Allows users to perform normal VSAM I/O (not improved control interval processing) to VSAM data sets.

- **CONTROL**  For VSAM data sets, CONTROL is equivalent to the VSAM CONTROL password; that is, it allows users to perform improved control interval processing—CONTROL is control-interval access (access to individual VSAM data blocks), and the ability to retrieve, update, insert, or delete records in the specified data set.

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

- **ALTER**  ALTER allows users to read, update, delete, rename, move, or scratch the data set.
When specified in a discrete profile, ALTER allows users to read, alter, and delete the profile itself *including the access list.*

ALTER does not allow users to change the owner of the profile using the ALTDSD command. However, if a user with ALTER access authority to a discrete data set profile renames the data set, changing the high-level qualifier to his or her own user ID, both the data set and the profile are renamed, and the OWNER of the profile is changed to the new user ID.

ALTER authority to a generic profile allows users to create new data sets that are covered by that profile, it does not give users authority over the profile itself.
Chapter 3. RACF TSO commands

Most RACF commands can be entered as RACF TSO commands. For a complete list of the RACF commands that can be entered as RACF TSO commands, see Table 2 on page 8.

RACF list commands, such as LISTUSER * or LISTGRP *, can generate many thousands of lines of output. This quantity of output is not very usable except as input to a processing program and can exhaust address space resources below the 16M line.

Restriction: RACF commands that provide output listings (LISTDSD, LISTGRP, LISTUSER, RACDCERT, RACLINK, RLIST and SETROPTS) are designed to be issued by users, not by programs. IBM does not support the processing of these commands by programs. The output format of these commands is not an intended interface and may change with any z/OS release or as the result of service (PTFs) applied within a release. Programs should not examine the output of these commands. Instead, programs should use documented programming interfaces, such as the following:

- The output file from the IRRDBU00 (database unload) utility, which was designed specifically for this use.
- The results returned by the RACROUTE REQUEST=EXTRACT request.
- The results returned by the ICHEINTY macro.
- The results returned by the _R_admin (IRRSEQ00) callable service when using one of the profile extract function codes, or one of the SETROPTS retrieval function codes.

The syntax of RACF TSO commands is the same as the syntax of TSO commands. For example, a comma or one or more blanks are valid delimiters for use between operands.

Notes:

1. "Syntax of RACF commands and operands" on page 9 contains the key to symbols used in the command syntax diagrams.

2. The TSO parse routines allow you to abbreviate an operand on a TSO command to the least number of characters that uniquely identify the operand. To avoid conflicts in abbreviations, it is a good practice to fully spell out all operands on commands that are hardcoded (as in programs and CLISTS, for example).

3. If you specify a keyword in the RACF segment multiple times on the same command, RACF uses only the last occurrence. If a keyword in a non-RACF segment (such as TSO, CICS, SESSION) is specified multiple times on the same command, the last occurrence is also used except for keywords where a list of values is valid (such as ADDUSER USER1 NETVIEW(OPCLASS(value1 value2 value3 ...))). For these keywords, all values on all specifications are accepted.

4. If you are sharing your RACF database with a VM system, you can administer VM classes from the MVS system. For RACF-provided VM classes see Supplied resource classes for z/VM systems. Detailed information about working with VM classes can be found in the for RACF 1.10 information.

5. Make sure your job or logon specifies a region size large enough to run the commands, or they may ABEND unpredictably.
How to enter RACF TSO commands

The following sections describe how to enter RACF TSO commands. You can enter RACF TSO commands directly in the foreground during a TSO terminal session or by using the RACF ISPF panels.

You can enter commands in the background by using a batch job. You cannot use alias data set names in the RACF commands or panels. Alias names are alternative names for a data set that are defined in the catalog. RACF does not allow alias names because RACF uses the RACF database, not the catalog, for its processing.

Choosing between Using RACF TSO Commands and ISPF Panels

In general, you can perform the same RACF functions using RACF TSO commands and ISPF panels.

The RACF TSO commands provide the following advantages:

- Entering commands can be faster than displaying many panels in sequence.
- Using commands from the documented examples is more straightforward. (The examples in the RACF documents are generally command examples.)
- Getting online help for RACF TSO commands
  You can get online help for the RACF TSO commands documented in z/OS Security Server RACF Command Language Reference.
  - To see online help for the PERMIT command, for example, enter:
  HELP PERMIT
  - To limit the information displayed, specify operands on the HELP command. For example, to see only the syntax of the PERMIT command, enter:
  HELP PERMIT SYNTAX
  Restriction: TSO online help is not available when RACF commands are entered as RACF operator commands.
- Getting message ID information
  If a RACF TSO command fails, you will receive a message. If you do not get a message ID, enter:
  PROFILE MSGID

  Reenter the RACF TSO command that failed. The message appears with the message ID. See z/OS Security Server RACF Messages and Codes for help if the message ID starts with ICH or IRR.
  Restriction: PROFILE MSGID cannot be entered as a RACF operator command.

The ISPF panels provide the following advantages:

- When you use the panels, you avoid having to memorize a command and type it correctly. Panels can be especially useful if the command is complex or you perform a task infrequently.
- ISPF creates in the ISPF log a summary record of the work that you do. Unless you use the TSO session manager, the RACF commands do not create such a record.
- From the panels, you can press the HELP key to display brief descriptions of the fields on the panels.
- The options chosen when installing the RACF panels determine whether output (for example, profile listings, search results, and RACF options) is displayed in a scrollable form.
The ISPF panels for working with password rules allow you to enter all of the password rules on one panel. Figure 2 shows one of these panels.

When you use the ISPF panels to update a custom field definition in the CFDEF segment, the current values are displayed. You can then overtype the values to make changes.

When you use the ISPF panels to add, update, or delete custom field information (CSDATA segment fields) in a user or group profile, the panels are primed with the custom field names and values. You can then make additions, changes, and deletions.

<table>
<thead>
<tr>
<th>RACF - SET PASSWORD FORMAT RULES</th>
</tr>
</thead>
<tbody>
<tr>
<td>COMMAND ==&gt;</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Enter PASSWORD FORMAT RULES:</td>
</tr>
<tr>
<td>MINIMUM_MAXIMUM</td>
</tr>
<tr>
<td>LENGTH LENGTH FORMAT</td>
</tr>
<tr>
<td>RULE 1: __ __ ________</td>
</tr>
<tr>
<td>RULE 2: __ __ ________</td>
</tr>
<tr>
<td>RULE 3: __ __ ________</td>
</tr>
<tr>
<td>RULE 4: __ __ ________</td>
</tr>
<tr>
<td>RULE 5: __ __ ________</td>
</tr>
<tr>
<td>RULE 6: __ __ ________</td>
</tr>
<tr>
<td>RULE 7: __ __ ________</td>
</tr>
<tr>
<td>RULE 8: __ __ ________</td>
</tr>
<tr>
<td>To cancel an existing rule, enter NO for MINIMUM LENGTH.</td>
</tr>
<tr>
<td>To specify FORMAT, use the following codes for each character position:</td>
</tr>
<tr>
<td>* = Any Character $ = National V = Vowel N = Numeric</td>
</tr>
<tr>
<td>C = Consonant A = Alphabetic v = Mixed Vowel m = Mixed Numeric</td>
</tr>
<tr>
<td>c = Mixed Consonant L = Alphanumeric w = No Vowel</td>
</tr>
</tbody>
</table>

Figure 2. Sample ISPF panel for RACF

Additional authorization for using the ISPF panels: You must authorize general users to use ISPF panels to add data to custom fields in user and group profiles. For details, see "Authorizing users to update data in custom fields" in z/OS Security Server RACF Security Administrator's Guide.

Restriction: The ISPF panels do not support all options of all commands. For example, the SETROPTS option to activate and deactivate mixed-case password support is not available through the RACF panels. For information about using the SETROPTS command to implement mixed-case passwords, see "Allowing Mixed-Case Passwords" in z/OS Security Server RACF Security Administrator's Guide.

Entering RACF TSO commands in the foreground

To enter RACF TSO commands in the foreground, you must be able to:
- Conduct a TSO terminal session
- Use the TSO commands
- Use system-provided aids (HELP command, attention interrupt, conversational messages)
- Respond to TSO prompts

See z/OS TSO/E Primer and z/OS TSO/E Command Reference for any information you need.

In addition, to enter RACF TSO commands from the foreground, you must be defined to the system.
TSO commands

The TSO LOGON command is used to define you to the system as a RACF user through the user identity (user ID), password, GROUP, and OIDCARD operands. To change your RACF password, use the new_password option on the LOGON command. If you have more than one account number defined in your TSO profile, you must supply an account number on the LOGON command.

The default data set name prefix in your TSO profile is used as the high-level qualifier of a DASD or tape data set name if you do not enter the fully qualified name in a TSO or RACF command. RACF also uses the TSO default prefix as the high-level qualifier for the name of a CLIST created as a result of the RACF SEARCH command. If you do not have a prefix specified in your TSO profile, (PROFILE NOPREFIX), the user ID from the SEARCH command issuer's ACEE is used as the qualifying prefix.

If you frequently use RACF TSO commands on RACF-protected data sets, you can set your TSO default prefix as follows:
• Set the default prefix to your user ID if you do a good deal of work with your own data sets.
• Set the default prefix to a specific RACF group name if you are working mostly with data sets from that group.

Although, the command examples in this document generally use uppercase characters, you can enter commands using either uppercase or lowercase characters.

Entering RACF TSO commands in the background

You can enter RACF TSO commands in the background by submitting a batch job as follows:
• Using the batch internal or remote reader facility of the job entry subsystem (JES)
• Using the TSO SUBMIT command from a terminal

The RACF data you need to enter on your JCL depends on whether the job entry subsystem (JES) at your installation includes the JES RACF user identification feature. If your level of JES includes the RACF user identification propagation feature, any jobs you submit to the background while logged onto TSO are automatically identified to RACF with the same user identifier. When the job runs, RACF uses your default group as your current connect group. (User, password, and group information is not required on the JOB statement, but if you do specify this information, it overrides the propagated specifications.)

If your level of JES does not include the RACF user identification propagation feature, you must include the USER, PASSWORD, and, optionally, GROUP parameters on the JCL JOB statement.

The USER, PASSWORD, and GROUP parameters on the JCL JOB statement identify you to the system as a RACF user. To change your RACF password, you can use the new-password operand of the PASSWORD command. For information on how to code these parameters, see z/OS MVS JCL Reference.

As an alternative to coding PASSWORD on JCL statements, you can use the TSO SUBMIT command (for systems that do not have the JES RACF user identification propagation feature) to automatically include this information during job submission. To use SUBMIT, you should code the USER (userid) and PASSWORD operands on...
the SUBMIT command. These operands are then put on the JCL JOB statement that the command generates. When the job runs, RACF uses the name of the user's default group as the current connect group.

Example of RACF TSO commands in the background
The following example shows how to submit RACF TSO commands in the background as a batch job:

```
//jobname JOB ...
//STEP1 EXEC PGM=IKJEFT01,DYNAMNBR=20
//SYSTSIN DD SYSOUT=A
//SYSTSIN DD *
ADDGROUP PROJECTA
ADDUER (PAJ5 ESH25)
ADDS 'PROJECTA.XYZ.DATA'
PERMIT 'PROJECTA.XYZ.DATA' ID(PAJ5) ACCESS(UPDATE)
/*
```

When a fully-qualified data set name is not given in a command entered in the background, the effect is the same as for a command entered in the foreground; the user's TSO default data set name prefix is used as the high-level qualifier of a DASD or tape data set name. The prefix is also used as the high-level qualifier for the name of a command procedure (CLIST) created as a result of the RACF SEARCH command. If the user is defined to TSO, the default prefix is in the TSO profile for the user specified in the USER parameter on the JCL JOB statement or the USER operand in the TSO SUBMIT command. If the user is not defined to TSO, there is no default prefix unless the TSO PROFILE command is used.
Chapter 4. RACF operator commands

The RACF operator commands allow you to perform RACF functions from an operator console. For a complete list of the RACF commands that can be entered as RACF operator commands, see Table 2 on page 8.

Note: Use of these commands requires that the RACF subsystem has been started. If the RACF subsystem has not been started at your installation, contact your system programmer.

These commands allow an MVS operator to perform certain RACF operations in the RACF subsystem. The RACF subsystem prefix in front of the command identifies the RACF subsystem as the processing environment. Some things to remember:

- RACF operator commands require an MVS environment with the RACF subsystem active.
- The DISPLAY and SIGOFF commands require APPC/MVS and persistent verification.
- If a command can be issued as both a RACF TSO command or a RACF operator command:
  - RACF first checks that the issuer is defined to RACF and if not, an error message is issued.
    If you are defined to RACF, RACF verifies that you have sufficient authority to the proper resource in the OPERCMDS class to determine if you have authority to issue the command as an operator command. See [z/OS Security Server RACF Security Administrator's Guide] for further information.
    If the OPERCMDS class is not active, or if no OPERCMDS profile exists, then the user is allowed to issue the command as a RACF operator command.
  - You must be logged on to the console to issue the command as a RACF operator command.

Note: The RVARY command is the exception, because it can be issued as both a RACF TSO command and an RACF operator command but does not have to fit these circumstances.

- If a command can be issued as a RACF operator command, but not as a RACF TSO command:
  - RACF first checks to see if you have OPERCMDS authority. If the user is logged on to the console, the OPERCMDS class is active, and a OPERCMDS profile exists, you have OPERCMDS authority.
  - If you are not logged on to the console, the OPERCMDS class is inactive, or no OPERCMDS profile exists, you can only issue the command from the master console or a console with system authority.

Note: The DISPLAY command is the exception, because it can be issued under these circumstances without any particular console authority.

Rules for entering RACF operator commands

1. A RACF operator command must contain the RACF subsystem prefix. A command such as the DISPLAY command could be entered on the command line as follows:

   #DISPLAY xxxx
Operator commands

Where:

# specifies the subsystem prefix. The subsystem prefix specifies that the RACF subsystem is the processing environment of the command. The subsystem prefix can be either the installation-defined prefix for RACF (1–8 characters) or, if no prefix has been defined, the RACF subsystem name followed by a blank. If the command prefix was registered with CPF, you can use the MVS command D OPDATA to display it or you can contact your system programmer.

If no subsystem prefix has been defined, and the subsystem name is rac1, the same command would be entered as follows:

```
racl DISPLAY xxxx
```

**Note:** If you need to find out what the subsystem prefix is, contact your system programmer.

`xxxx` specifies DISPLAY operands.

2. Separate operands with commas. Do not specify commas before the first operand or after the last operand.

   For example, enter a DISPLAY command with two operands as follows:

   ```
   #DISPLAY xxxx,yyyy
   ```

3. You can also separate operands with blanks. This practice is not encouraged, however, because future releases might not allow this.

4. The order in which you specify the operands on the command line does not affect the command. For example: the commands `#DISPLAY xxxx,yyyy` and `#DISPLAY yyyy,xxxx` give the same result.

5. RACF commands entered as RACF operator commands must meet the MVS restrictions on command length and operand content. A command with intended mixed-case input cannot be entered as an operator command since it is automatically translated to uppercase before it is sent to RACF. In addition, command messages and output from RACF to the console are also translated to uppercase. When dealing with profile names or fields requiring mixed-case characters, enter the command as a TSO command, not as an operator command.
Chapter 5. RACF command syntax

This topic describes the syntax and function of the RACF commands. The commands are presented in alphabetic order. Each command description contains several examples.
ADDGROUP

ADDGROUP (Add group profile)

Purpose

Use the ADDGROUP command to define a new group to RACF.

The command adds a profile for the new group to the RACF database. It also establishes the relationship of the new group to the superior group you specify.

Group profiles consist of a RACF segment and, optionally, other segments such as DFP and OMVS. You can use this command to specify information in any segment of the profile.

Issuing options

The following table identifies the eligible options for issuing the ADDGROUP command:

<table>
<thead>
<tr>
<th>As a RACF TSO command?</th>
<th>As a RACF operator command?</th>
<th>With command direction?</th>
<th>With automatic command direction?</th>
<th>From the RACF parameter library?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

For information on issuing this command as a RACF TSO command, refer to Chapter 3, “RACF TSO commands,” on page 15.

For information on issuing this command as a RACF operator command, refer to Chapter 4, “RACF operator commands,” on page 21.

You must be logged on to the console to issue this command as a RACF operator command.

Related commands

- To delete a group profile, see DELGROUP (Delete group profile).
- To change a group profile, see ALTGROUP (Alter group profile).
- To connect a user to a group, see CONNECT (Connect user to group).
- To remove a user from a group, see REMOVE (Remove user from group).
- To obtain a list of group profiles, see SEARCH (Search RACF database).
- To list a group profile, see LISTGRP (List group profile).

Authorization required

To use the ADDGROUP command, you must have at least one of the following authorizations:

- Have the SPECIAL attribute,
- Have the group-SPECIAL attribute and the superior group is within your group-SPECIAL scope,
- Be the owner of the superior group, or
- Have JOIN authority in the superior group.

When issuing the ADDGROUP command as a RACF operator command, you might require sufficient authority to the proper resource in the OPERCMDS class. See z/OS Security Server RACF Security Administrator's Guide for further information.
To add segments, such as DFP or OMVS, to a group’s profile, you must have at least one of the following authorizations:

- You must have the SPECIAL attribute.
- Your installation must permit you to do so through field-level access checking.

For information on field-level access checking, see *z/OS Security Server RACF Security Administrator’s Guide*.

To specify the AT keyword, you must have READ authority to the DIRECT.node resource in the RRSFDATA class and a user ID association must be established between the specified node.userid pair(s).

To specify the ONLYAT keyword you must have the SPECIAL attribute, the userid specified on the ONLYAT keyword must have the SPECIAL attribute, and a user ID association must be established between the specified node.userid pair(s) if the user IDs are not identical.

To specify the SHARED keyword, you must have the SPECIAL attribute or at least READ authority to the SHARED.IDS resource in the UNIXPRIV class.

**Syntax**

For the key to the symbols used in the command syntax diagrams, see “Syntax of RACF commands and operands” on page 9. The complete syntax of the ADDGROUP command is:

```plaintext
[subsystem-prefix]ADDGROUP | AG
   (group-name ...)
   [ AT([(node).userid ...] | ONLYAT([(node).userid ...])]
   [ CSDATA(
       [ custom-field-name(custom-field-value) ] ...
    )]
   [ DATA('installation-defined-data')]
   [ DFP(
       [ DATAAPPL(application-name) ]
       [ DATACLAS(data-class-name) ]
       [ MGMTCLAS(management-class-name) ]
       [ STORAGECLAS(storage-class-name) ]
    )]
   [ MODEL(dsname) ]
   [ OMVS
      [ ( AUTOGID | GID(group-identifier) [ SHARED ] ) ] ]
   [ OVM
      [ ( GID(group-identifier) ) ]
      [ OWNER(userid or group-name) ]
      [ SUPGROUP(group-name) ]
      [ TERMUACC | NOTERMUACC ]
      [ TME(
          [ ROLES(profile-name ...)] ) ]
      [ UNIVERSAL ]
```

For information on issuing this command as a RACF TSO command, refer to Chapter 3, “RACF TSO commands,” on page 15.

For information on issuing this command as a RACF operator command, refer to Chapter 4, “RACF operator commands,” on page 21.
ADDGROUP

Parameters

subsystem-prefix
Specifies that the RACF subsystem is the processing environment of the command. The subsystem prefix can be either the installation-defined prefix for RACF (1–8 characters) or, if no prefix has been defined, the RACF subsystem name followed by a blank. If the command prefix was registered with CPF, you can use the MVS command D OPDATA to display it or you can contact your RACF security administrator.

Only specify the subsystem prefix when issuing this command as a RACF operator command. The subsystem prefix is required when issuing RACF operator commands.

group-name
Specifies the name of the group whose profile is to be added to the RACF database. If you are defining more than one group, the list of group names must be enclosed in parentheses.

This operand is required and must be the first operand following ADDGROUP. Each group-name must be unique and must not currently exist in the RACF database as a group name or a user ID.

AT | ONLYAT
The AT and ONLYAT keywords are only valid when the command is issued as a RACF TSO command.

AT([node].userid ...) Specifies that the command is to be directed to the node specified by node, where it runs under the authority of the user specified by userid in the RACF subsystem address space.

If node is not specified, the command is directed to the local node.

ONLYAT([node].userid ...) Specifies that the command is to be directed only to the node specified by node where it runs under the authority of the user specified by userid in the RACF subsystem address space.

If node is not specified, the command is directed only to the local node.

CSDATA
Specifies information to add a custom field for this group.

Usage for each custom field is defined using the CFDEF operand of the RDEFINE command for resource profiles in the CFIELD class. Contact your security administrator to see how custom fields are used at your installation. For more information about custom fields, see z/OS Security Server RACF Security Administrator's Guide.

custom-field-name(custom-field-value) ...
Specifies the name and value of a custom field for this group. You can add values for multiple custom field values with a single ADDGROUP command.

Rules:
- You must use the same custom-field-name as defined by the CFIELD profile named GROUP:CSDATA:custom-field-name. (The CFIELD profile is defined using the CFDEF operand of the RDEFINE command.)
- You must specify a custom-field-value that is valid for the attributes of this custom field. (The attributes, such as data type, are defined in the CFDEF segment of the CFIELD profile.)
ADDGROUP

DATA('installation-defined-data')
Specifies up to 255 characters of installation-defined data to be stored in the
group profile and must be enclosed in single quotation marks. It might also
contain double-byte character set (DBCS) data.

Use the LISTGRP command to list this information.

DFP
Specifies that when you define a group to RACF, you can enter any of the
following suboperands to specify default values for the DFP data, management,
and storage classes. DFP uses this information to determine data management
and DASD storage characteristics when a user creates a new group data set.

DATAAPPL(application-name)
Specifies DFP data application identifier. The maximum length of a data
class name is 8 characters.

DATACLAS(data-class-name)
Specifies the default data class. The maximum length of a data class name
is 8 characters.

A data class can specify some or all of the physical data set attributes
associated with a new data set. During new data set allocation, data
management uses the value you specify as a default unless it is preempted
by a higher priority default, or overridden in some other way (for example,
by JCL).

Note: The value you specify must be a valid data class name defined for
use on your system. For more information, see z/OS Security Server
RACF Security Administrator’s Guide
For information on defining DFP data classes, see z/OS DFSMSdfp Storage
Administration.

MGMTCLAS(management-class-name)
Specifies the default management class. The maximum length of a
management class name is 8 characters.

A management class contains a collection of management policies that
apply to data sets. Data management uses the value you specify as a
default unless it is preempted by a higher priority default, or overridden in
some other way (for example, by JCL).

Note: The value you specify must be defined as a profile in the
MGMTCLAS general resource class, and the group must be granted
at least READ access to the profile. Otherwise, RACF does not allow
the group access to the specified MGMTCLAS. For more
information, see z/OS Security Server RACF Security Administrator’s
Guide
For information on defining DFP management classes, see z/OS
DFSMSdfp Storage Administration.

STORCLAS(storage-class-name)
Specifies the default storage class. The maximum length of a
storage-class-name is 8 characters.

A storage class specifies the service level (performance and availability) for
data sets managed by the Storage Management Subsystem (SMS). During
new data set allocation, data management uses the value you specify as a
default unless it is preempted by a higher priority default, or overridden in some other way (for example, by JCL).

Note: The value you specify must be defined as a profile in the STORCLAS general resource class, and the group must be granted at least READ access to the profile. Otherwise, RACF does not allow the group access to the specified STORCLAS. For more information, see z/OS Security Server RACF Security Administrator’s Guide.

For information on defining DFP storage classes, see z/OS DFSMSdfp Storage Administration.

MODEL(dsname)
Specifies the name of a discrete MVS data set profile to be used as a model for new group-name data sets. For this operand to be effective, the MODEL(GROUP) option (specified on the SETROPTS command) must be active.

RACF always prefixes the data set name with group-name when it accesses the model.

For information about automatic profile modeling, refer to z/OS Security Server RACF Security Administrator’s Guide.

OMVS
Specifies z/OS UNIX System Services information for the group being defined to RACF.

AUTOGID | GID
Specifies whether RACF is to automatically assign an unused GID value to the group or if a specific GID value is to be assigned.

AUTOGID
Specifies that RACF is to automatically assign an unused GID value to the group. The GID value is derived from information obtained from the BPX.NEXT.USER profile in the FACILITY class. For more information on setting up BPX.NEXT.USER, see z/OS Security Server RACF Security Administrator’s Guide.

If you are using RRSF automatic command direction for the GROUP class, the command sent to other nodes will contain an explicit assignment of the GID value which was derived by RACF on the local node.

Rules:
• AUTOGID cannot be specified if more than one group is entered.
• The AUTOGID keyword is mutually exclusive with the SHARED keyword.
• If both GID and AUTOGID are specified, AUTOGID is ignored.
• Field-level access checking for the GID field applies when using AUTOGID.

GID(group-identifier) [SHARED]
GID(group-identifier)
Specifies the group identifier. The GID is a numeric value from 0–2 147 483 647.

When a GID is assigned to a group, all users connected to that group who have a user identifier (UID) in their user profile can use
functions such as the TSO/E command, OMVS, and can access z/OS UNIX files based on the GID and UID values assigned.

**Rules:**
- If the security administrator has defined the SHARED.IDS profile in the UNIXPRIV class, the GID must be unique. Use the SHARED keyword in addition to GID to assign a value that is already in use.
- If SHARED.IDS is not defined, RACF does not require the GID to be unique. The same value can be assigned to multiple groups, but this is not recommended because individual group control would be lost. However, if you want a set of groups to have exactly the same access to z/OS UNIX resources, you might decide to assign the same GID to more than one group.
- RACF allows you to define and connect a user to more than 300 (which is the same as the NGROUPS_MAX variable defined in the POSIX standard) groups, but when a process is created or z/OS UNIX group information is requested, only up to the first 300 z/OS UNIX groups are associated with the process or user. The first 300 z/OS UNIX groups, that have GIDs, to which a user is connected are used by z/OS UNIX. LISTUSER displays the groups in the order that RACF examines them when determining which of the user's groups are z/OS UNIX groups.

See [z/OS UNIX System Services Planning](#) for information on NGROUPS_MAX.

**SHARED**

If the security administrator has chosen to control the use of shared GIDs, this keyword must be used in addition to the GID keyword to specify the group identifier if it is already in use by at least one other group. The administrator controls shared GIDs by defining the SHARED.IDS profile in the UNIXPRIV class.

**Rules:**
- If the SHARED.IDS profile is not defined, SHARED is ignored.
- If SHARED is specified in the absence of GID, it is ignored.
- If the SHARED.IDS profile is defined and SHARED is specified, but the value specified with GID is not currently in use, SHARED is ignored and UNIXPRIV authority is not required.
- Field-level access checking for the GID field applies when using SHARED.
- The SHARED keyword is mutually exclusive with the AUTOGID keyword.

**OVM**

Specifies OpenExtensions VM information for the group being defined to RACF.

**GID(group-identifier)**

Specifies the OpenExtensions VM group identifier. The GID is a numeric value from 0–2 147 483 647.

If you do not specify GID, the group is assigned the default GID of 4294967295 (X'FFFFFFFF') and a LISTGRP shows NONE for the GID.
**ADDGROUP**

**Notes:**

1. RACF does not require the GID to be unique. The same value can be assigned to multiple groups, but this is not recommended because individual group control would be lost. However, if you want a set of groups to have exactly the same access to the OpenExtensions VM resources, you might decide to assign the same GID to more than one group.

2. The value defined for the NGROUPS_MAX variable in the ICHNGMAX macro on VM defines the maximum number of OpenExtensions VM groups to be associated with an OpenExtensions VM process or user. The NGROUPS_MAX variable on VM is a number from 32–125, inclusive. However, RACF allows you to define and connect a user to more than the number of groups defined in this variable. If the NGROUPS_MAX variable is \( n \) and a process is created or OpenExtensions VM group information is requested, only up to the first \( n \) OpenExtensions VM groups are associated with the process or user. The first \( n \) OpenExtensions VM groups to which a user is connected are used by OpenExtensions VM. LISTUSER displays the groups in the order that RACF examines them when determining which of the user’s groups are OpenExtensions VM groups.

   See [z/OS Security Server RACF Macros and Interfaces](https://www.ibm.com/support/knowledgecenter/SSS79A_11.1.0/com.ibm.zos.racf.doc/racmics_ichngmax.htm) for information on NGROUPS_MAX.

**OWNER(userid or group-name)**

Specifies a RACF-defined user or group to be assigned as the owner of the new group. If you do not specify an owner, you are defined as the owner of the group. If you specify a group name, it must be the name of the superior group for the group you are adding.

**SUPGROUP(group-name)**

Specifies the name of an existing RACF-defined group. This group becomes the superior group of the group profile you are defining.

If you omit SUPGROUP, RACF uses your current connect group as the superior group.

If you specify a group name and also specify OWNER with a group name, you must use the same group name on both SUPGROUP and OWNER.

If your authority to issue ADDGROUP comes from the group-SPECIAL attribute, any group you specify must be within the scope of the group in which you are a group-SPECIAL user.

**TERMUACC | NOTERMUACC**

**TERMUACC**

Specifies that during terminal authorization checking, RACF allows any user in the group access to a terminal based on the universal access authority for that terminal. TERMUACC is the default value if you omit both TERMUACC and NOTERMUACC.

**NOTERMUACC**

Specifies that the group or a user connected to the group must be explicitly authorized (through the PERMIT command with at least READ authority) to access a terminal.

**TME**

Specifies that information for the Tivoli® Security Management Application is to be added.
ADDGROUP

Note: The TME segment fields are intended to be updated only by the Tivoli Security Management Application, which manages updates, permissions, and cross references. A security administrator should only directly update Tivoli Security Management fields on an exception basis.

ROLES(profile-name ...)
Specifies a list of roles that reference this group.

The profile-name value should be the name of a defined role, which is a discrete general resource profile in the ROLE class.

UNIVERSAL
Specifies that this is a universal group that allows an effectively unlimited number of users to be connected to it for the purpose of resource access. The number of users in a universal group with authority higher than USE, or with the attributes SPECIAL, OPERATIONS or AUDITOR at the group level, is still limited to 5957.

When displayed with the LISTGRP command, not all group members will be listed. Only users with authority higher than USE or with the attributes SPECIAL, OPERATIONS or AUDITOR at the group level will be shown in the member list.

Examples

Table 3. ADDGROUP examples

Example 1

Operation User IA0 wants to add the group PROJECTA as a subgroup of RESEARCH. User IA0 will be the owner of group PROJECTA. Users in group PROJECTA will be allowed to access a terminal based on the universal access authority assigned to that terminal.

Known User IA0 has JOIN authority to group RESEARCH. User IA0 is currently connected to group RESEARCH. User IA0 wants to issue the command as a RACF operator command, and the RACF subsystem prefix is @.

Command @ADDGROUP PROJECTA

Defaults SUPGROUP(RESEARCH) OWNER(IA0) TERMUACC

Example 2

Operation User ADM1 wants to add the group PROJECTB as a subgroup of RESEARCH. Group RESEARCH will be the owner of group PROJECTB. Group PROJECTB must be authorized to use terminals through the PERMIT command.

Known User ADM1 has JOIN authority to group RESEARCH. User ADM1 is currently connected to group SYS1. USER ADM1 wants to issue the command as a RACF TSO command.

Command ADDGROUP PROJECTB SUPGROUP(RESEARCH) OWNER(RESEARCH) NOTERMUACC

Defaults None.

Example 3

Operation User ADM1 wants to add the group SYSINV as a subgroup of RESEARCH. This group will be used as the administrative group for RACF and will use a model name of SYSINV.RACF.MODEL.PROFILE. User ADM1 wants to direct the command to run under the authority of user APW02.

Known User APW02 has JOIN authority to group RESEARCH and ADM1 wants to issue the command as a RACF TSO command.

ADM1 and APW02 have an established user ID association.

Command ADDGROUP SYSINV SUPGROUP(RESEARCH) MODEL(RACF.MODEL.PROFILE) DATA('RACF ADMINISTRATION GROUP') AT(.APW02)

Defaults OWNER(ADM1) TERMUACC

Command direction defaults to the local node.
ADDGROUP

Table 3. ADDGROUP examples (continued)

Example 4

**Operation**
User ADM1 wants to add the group DFPADMN as a subgroup of SYSADMN. Group SYSADMN will be the owner of group DFPADMN. Users in group DFPADMN will be allowed to access a terminal based on the universal access authority assigned to that terminal. Group DFPADMN will be assigned the following default information to be used for new DFP-managed data sets created for the group:

- Data class DFP2DATA
- Management class DFP2MGMT
- Storage class DFP2STOR
- Data application identifier DFP2APPL.

**Known**
- User ADM1 has JOIN authority to group SYSADMN.
- User ADM1 is currently connected to group SYS1.
- User ADM1 has field-level access of ALTER to the fields in the DFP segment.
- DFP2MGMT has been defined to RACF as a profile in the MGMTCLAS general resource class, and group DFPADMN has been given READ access to this profile.
- DFP2STOR has been defined to RACF as a profile in the STORCLAS general resource class, and group DFPADMN has been given READ access to this profile.
- User ADM1 wants to issue the command as a RACF TSO command.

**Command**
```
ADDGROUP DFPADMN SUPGROUP(SYSADMN) OWNER(SYSADMN)
DFP(DATACLAS(DFP2DATA) MGMTCLAS(DFP2MGMT) STORCLAS(DFP2STOR)
DATAAPPL(DFP2APPL))
```

**Defaults**
TERMUACC

Example 5

**Operation**
User ADM1 wants to add the UNIVERSAL group NETGROUP as a subgroup of SYS1. User IBMUSER will be the owner of group NETGROUP. Universal group NETGROUP can have an unlimited number of members (that have USE authority and are not SPECIAL, OPERATIONS, or AUDITOR).

**Known**
- User ADM1 has group-SPECIAL authority to group SYS1.
- User ADM1 is currently connected to group SYS1.

**Command**
```
ADDGROUP NETGROUP DATA('INTERNET CUSTOMER GROUP') SUPGROUP(SYS1)
OWNER(IBMUSER) UNIVERSAL
```

**Defaults**
None apply.

Example 6

**Operation**
User RACFADM with SPECIAL or UPDATE authority requests the addition of a new z/OS UNIX group. The user specifies AUTOGID so that RACF will automatically assign an unused GID to the new user.

**Known**
The group profile is owned by RACFADM and belongs to RACFADM's current connect group SYSOM. The BPX.NEXT.USER profile in the FACILITY class has been set up to allow automatic GID assignment.

**Command**
```
ADDGROUP UNIXGRP OMVS(AUTOGID HOME('/u/unixgrp') CPUTIMEMAX(5000)
ASSIZEMAX(40000000))
```

**Defaults**
DFLTGRP(SYSOM) OWNER(RACFADM)
ADDSD (Add data set profile)

Purpose

Use the ADDSD command to add RACF protection to data sets with either discrete or generic profiles.

Changes made to discrete profiles take effect after the ADDSD command is processed. Changes made to generic profiles do not take effect until one or more of the following steps is taken:

- The user of the data set issues the LISTDSD command:
  
  `LISTDSD DA(data-set-protected-by-the-profile) GENERIC`

  Note: Use the data set name, not the profile name.

- The security administrator issues the SETROPTS command:
  
  `SETROPTS GENERIC(DATASET) REFRESH`

  See SETROPTS command for authorization requirements.

- The user of the data set logs off and logs on again.


Issuing options

The following table identifies the eligible options for issuing the ADDSD command:

<table>
<thead>
<tr>
<th>As a RACF TSO command?</th>
<th>As a RACF operator command?</th>
<th>With command direction?</th>
<th>With automatic command direction?</th>
<th>From the RACF parameter library?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

For information on issuing this command as a RACF TSO command, refer to Chapter 3, “RACF TSO commands,” on page 15.

For information on issuing this command as a RACF operator command, refer to Chapter 4, “RACF operator commands,” on page 21.

You must be logged on to the console to issue this command as a RACF operator command.

Related commands

- To delete a data set profile, see [DELDSD (Delete data set profile)](http://www.ibm.com/support/docview.wss?rs=1366&context=SSIG&context=SSIG&docid=SSIG/deldsd).
- To permit or deny access to a data set profile, see [PERMIT (Maintain resource access lists)](http://www.ibm.com/support/docview.wss?rs=1366&context=SSIG&context=SSIG&docid=SSIG/permit).
- To obtain a list of data set profiles, see [SEARCH (Search RACF database)](http://www.ibm.com/support/docview.wss?rs=1366&context=SSIG&context=SSIG&docid=SSIG/search).
- To list a data set profile, see [LISTDSD (List data set profile)](http://www.ibm.com/support/docview.wss?rs=1366&context=SSIG&context=SSIG&docid=SSIG/list).

Authorization required

When issuing the ADDSD command as a RACF operator command, you might require sufficient authority to the proper resource in the OPERCMDS class. See [z/OS Security Server RACF Security Administrator’s Guide](http://www.ibm.com/support/docview.wss?rs=1366&context=SSIG&context=SSIG&docid=SSIG/guide) for further information.
**ADDSD**

**Notes:**
1. You need not have the SPECIAL attribute to specify the OWNER operand.
2. To specify the AT keyword, you must have READ authority to the DIRECT.node resource in the RRSFDATA class and a user ID association must be established between the specified node.userid pair(s).
3. To specify the ONLYAT keyword you must have the SPECIAL attribute, the userid specified on the ONLYAT keyword must have the SPECIAL attribute, and a user ID association must be established between the specified node.userid pair(s) if the user IDs are not identical.

The level of authority you need to use the ADDSD command and the types of profiles you can define are:

- **To protect a user data set with RACF, one of the following must be true:**
  - The high-level qualifier of the data set name (or the qualifier supplied by the RACF naming conventions table or by a command installation exit) must match your user ID.
  - You must have the SPECIAL attribute.
  - The user ID for the data set profile must be within the scope of a group in which you have the group-SPECIAL attribute.

- **To protect a group data set with RACF, one of the following must be true:**
  - You must have at least CREATE authority in the group.
  - You must have the SPECIAL attribute.
  - You must have the OPERATIONS attribute and not be connected to the group.
  - The data set profile must be within the scope of the group in which you have the group-SPECIAL attribute.
  - The data set profile must be within the scope of the group in which you have the group-OPERATIONS attribute, and you must not be connected to the group.

**Notes:**
1. If you have the OPERATIONS or group-OPERATIONS attribute and are connected to a group, you must have at least CREATE authority in that group to protect a group data set.
2. When creating a group data set profile, the profile creator’s user ID is placed on the access list with ALTER authority unless the creation was allowed due to OPERATIONS or group-OPERATIONS authority or unless the SETROPTS NOADDCREATOR option is in effect.

- To define to RACF a data set that was brought from another system where it was RACF-indicated and RACF-protected with a discrete profile, one of the following must be true:
  - You must either have the SPECIAL attribute, or the data set’s profile is within the scope of a group in which you have the group-SPECIAL attribute
  - Your user ID must be the high-level qualifier of the data set name (or the qualifier supplied by the naming conventions routine or a command installation exit).

- To assign a security category to a profile, you must have the SPECIAL attribute or have the category in your user profile.
- To assign a security level to a profile, you must have the SPECIAL attribute or, in your own profile, a security level that is equal to or greater than the security level you are defining.
To assign a security label to a profile, you must have the SPECIAL attribute or READ authority to the security label profile. However, the security administrator can limit the ability to assign security labels to only users with the SPECIAL attribute.

To access the DFP or TME segment, field-level access checking is required.

When either a user or group uses modeling to protect a data set with a discrete profile, RACF copies the following fields from the model profile: the level number, audit flags, global audit flags, the universal access authority (UACC), the owner, the warning, the access list, installation data, security category names, the security level name, the user to be notified, the retention period for a tape data set, and the erase indicator.

To add a discrete profile for a VSAM data set already RACF-protected by a generic profile, you must have ALTER access authority to the catalog or to the data set through the generic profile.

**Model profiles:** To specify a model data set profile (using, as required, FROM, FCLASS, FGENERIC, and FVOLUME), you must have sufficient authority over the model profile (the from profile). RACF makes the following checks until one of the conditions is met:

- You have the SPECIAL attribute.
- The from profile is within the scope of a group in which you have the group-SPECIAL attribute.
- You are the owner of the from profile.
- The high-level qualifier of the profile name (or the qualifier supplied by the naming conventions routine or a command installation exit routine) is your user ID.
- For a discrete profile, you are on the access list in the from profile with ALTER authority. (If you have any lower level of authority, you cannot use the profile as a model.)
- For a discrete profile, your current connect group (or, if list-of-groups checking is active, any group to which you are connected) is on the access list in the from profile with ALTER authority.
- For a discrete profile, the UACC is ALTER.

### Syntax

For the key to the symbols used in the command syntax diagrams, see “Syntax of RACF commands and operands” on page 9. The complete syntax of the ADDSD command is:

```plaintext
[subsystem-prefix][ADDSD | AD]

(profile-name-1 [(/password] ...)

[ ADDCATEGORY(category-name ...)]
[ AT([node].userid ...] | ONLYAT([node].userid ...)]
[ AUDIT(access-attempt[(audit-access-level)] ...)]
[ DATA('installation-defined-data']
[ DFP(RESOWNER(userid or group-name) | NORESOWNER)]
[ ERASE ]
[ FCLASS(profile-name-2-class)]
[ FGENERIC ]
[ FILESEQ(number)]
[ FROM(profile-name-2)]
[ FVOLUME(profile-name-2-serial)]
```
Parameters

**subsystem-prefix**
Specifies that the RACF subsystem is the processing environment of the command. The subsystem prefix can be either the installation-defined prefix for RACF (1–8 characters) or, if no prefix has been defined, the RACF subsystem name followed by a blank. If the command prefix was registered with CPF, you can use the MVS command D OPDATA to display it or you can contact your RACF security administrator.

Only specify the subsystem prefix when issuing this command as a RACF operator command. The subsystem prefix is required when issuing RACF operator commands.

**profile-name-1**
Specifies the name of the discrete or generic profile to be added to the RACF database. If you specify more than one name, the list of names must be enclosed in parentheses.

The format of the profile name should follow the TSO/E data set naming conventions, except that the high-level qualifier of the profile name (or the qualifier determined by the naming conventions table or by a command installation exit) must be a user ID or a group name. See [z/OS Security Server RACF Security Administrator's Guide](https://publib.boulder.ibm.com/infocenter/zos/v2r11/index.jsp?topic=/com.ibm.zos.v2r11.doc/rassec_1b09.htm) for more information about the TSO/E data set naming conventions.

To specify a user ID other than your own, you must have the SPECIAL attribute, or the data set profile must be within the scope of a group in which you have the group-SPECIAL attribute. To define a group data set, you must have at least CREATE authority in the specified group, or the SPECIAL attribute, or the data set must be within the scope of a group in which you have the group-SPECIAL attribute.

This operand is required and must be the first operand following ADDSD. Note that, because RACF uses the RACF database and not the system catalog, you cannot use alias data set names.
ADDSD

For additional information, see "Profile names for data sets" on page 648 and the section describing rules for defining data set profiles in \[z/OS Security Server RACF Security Administrator's Guide\].

**Tape data set:** If you are defining a discrete profile that protects a tape data set, you must specify TAPE. If you are defining more than one tape data set profile, the data sets must all reside on the same volume, and you must specify the profile names in an order that corresponds to the file sequence numbers of the data sets on the volume.

**VSAM data set:** All of the components of a VSAM data set are protected by the profile that protects the cluster name. It is not necessary to create profiles that protect the index and the data components of the cluster.

**Data sets cataloged by an indirect VOLSER:** When you catalog a data set using an *indirect* VOLSER—using asterisks (******) or a symbolic such as &SYSRS in place of the VOLSER—you can protect the data set with a generic profile (preferred method) or else with one or more discrete data set profiles that contain the real unit and volume for each data set covered by the catalog entry. The latter must be done while the data set is online.

**/password**

Specifies the data set password if you are protecting an existing password-protected data set. If you specify a generic or model profile, RACF ignores this operand.

For a non-VSAM password-protected data set, the WRITE level password must be specified.

For a VSAM data set that is not password-protected, you do not need the password or RACF access authority for the catalog.

A password is not required when you specify NOSET.

If the command is executing in the foreground and you omit the password for a password-protected data set, the logon password is used. You are prompted if the password you enter or the logon password is incorrect. (If it is a non-VSAM multivolume data set, you are prompted once for each volume on which the data set resides.)

If the command is executing in a batch job and you either omit the password for a password-protected data set or supply an incorrect password, the operator is prompted. (If it is a non-VSAM multivolume data set, the operator is prompted once for each volume on which the data set resides.)

**ADDCATEGORY\((category-name ...))\)**

Specifies one or more names of installation-defined security categories. The names you specify must be defined as members of the CATEGORY profile in SECDATA class. (For information on defining security categories, see \[z/OS Security Server RACF Security Administrator's Guide\].)

When the SECDATA class is active and you specify ADDCATEGORY, RACF performs security category checking in addition to its other authorization checking. If a user requests access to a data set, RACF compares the list of security categories in the user's profile with the list of security categories in the data set profile. If RACF finds any security category in the data set profile that is not in the user's profile, RACF denies access to the data set. If the user's profile contains all the required security categories, RACF continues with other authorization checking.
Note: RACF does not perform security category checking for a started task or user that has the RACF privileged or trusted attribute. The RACF privileged or trusted attribute can be assigned to a started task through the RACF started procedures table or STARTED class, or to other users by installation-supplied RACF exits.

AT | ONLYAT
The AT and ONLYAT keywords are only valid when the command is issued as a RACF TSO command.

AT([node].userid ...)
Specifies that the command is to be directed to the node specified by node, where it runs under the authority of the user specified by userid in the RACF subsystem address space.

If node is not specified, the command is directed to the local node.

ONLYAT([node].userid ...)
Specifies that the command is to be directed only to the node specified by node where it runs under the authority of the user specified by userid in the RACF subsystem address space.

If node is not specified, the command is directed only to the local node.

AUDIT(access-attempt[ (audit-access-level)])...
Specifies which access attempts and access levels you want logged to the SMF data set.

access-attempt
Specifies which access attempts you want logged to the SMF data set. The following options are available:

ALL Specifies that you want to log both authorized accesses and detected unauthorized access attempts.

FAILURES Specifies that you want to log detected unauthorized attempts. FAILURES is the default value if you omit access-attempt.

NONE Specifies that you do not want any logging to be done.

SUCCESS Specifies that you want to log authorized accesses.

audit-access-level
Specifies which access levels you want logged to the SMF data set. The levels you can specify are:

ALTER Logs ALTER access-level attempts only.

CONTROL Logs access attempts at the CONTROL and ALTER levels.

READ Logs access attempts at any level. READ is the default value if you omit audit-access-level.

UPDATE Logs access attempts at the UPDATE, CONTROL, and ALTER levels.

FAILURES(READ) is the default value if you omit the AUDIT operand. You cannot audit access attempts at the EXECUTE level.

DATA('installation-defined-data')
Specifies up to 255 characters of installation-defined data to be stored in the data set profile and must be enclosed in single quotation marks. It might also contain double-byte character set (DBCS) data.
ADDSD

Use the LISTDSD command to list this information.

DFP
Specifies that for an SMS-managed data set, you can enter the following information:

RESOWNER(userid or group-name) | NORESOWNER
Specifies the user ID or group of the actual owner of the data sets protected by the profile specified in profile-name-1. This name must be that of a RACF-defined user or group. (The data set resource owner, specified with RESOWNER, is distinguished from the owner specified with OWNER, which represents the user or group that owns the data set profile).

If NORESOWNER is specified, the user or group represented by the high level qualifier of the data set profile is assigned as the owner of data sets protected by the profile when SMS needs to determine the RESOWNER.

ERASE
Specifies that when SETROPTS ERASE is active, data management is to physically erase the contents of deleted data sets and scratched or released DASD extents. Erasing the data set means overwriting its contents with binary zeroes so that it cannot be read.

Restrictions: The ERASE operand is ignored when any of the following conditions exist:

- When the data set is a tape data set and your installation did not activate the TAPEAUTHDSN option in the DEVSUPxx member of SYS1.PARMLIB. See “Erasing Scratched or Release Data (ERASE Option)” in z/OS Security Server RACF Security Administrator’s Guide for more information.
- When SETROPTS NOERASE is active for your installation. (User and data set profile definitions are overridden.)

FCLASS(profile-name-2-class)
Specifies the name of the class to which profile-name-2 belongs. The valid class names are DATASET and those classes defined in the class descriptor table. If you omit this operand, RACF assumes the DATASET class. This operand is valid only when you also specify the FROM operand; otherwise, RACF ignores it.

FGENERIC
Specifies that RACF is to treat profile-name-2 as a generic name, even if it is fully qualified (meaning that it does not contain any generic characters). This operand is only needed when profile-name-2 is a DATASET profile.

FILESEQ(number)
Specifies the file sequence number for a tape data set. The number can range from 1 through 65535.

If you specify more than one profile name, RACF assigns the file sequence number that you specify to the first profile name, then increments the number by one for each additional name. Thus, be sure to specify profile names in the order of their file sequence numbers.

If you omit FILESEQ, the default is FILESEQ(1). If you omit VOLUME, RACF retrieves the volume serial number from the catalog.

If you omit TAPE, RACF ignores FILESEQ.

FROM(profile-name-2)
Specifies the name of an existing discrete or generic profile that RACF is to use as a model for the new profile. The model profile name you specify on the
FROM operand overrides any model name specified in your user or group profile. If you specify FROM and omit FCLASS, RACF assumes that profile-name-2 is the name of a profile in the DATASET class.

To specify FROM, you must have sufficient authority to both profile-name-1 and profile-name-2, as described in [Authorization required].

Naming conventions processing affects profile-name-2 in the same way that it affects profile-name-1.

Mixed-case profile names are accepted and preserved when FCLASS refers to a class defined in the static class descriptor table with CASE=ASIS or in the dynamic class descriptor table with CASE(ASIS).

If the profile being added is for a group data set and the user has the GRPACC attribute for that group, RACF places the group on the access list with UPDATE access authority. Otherwise, if the group is already on the access list, RACF changes the group's access authority to UPDATE.

Possible Changes to Copied Profiles When Modeling Occurs

When a profile is copied during profile modeling, the new profile could differ from the model in the following ways:

- Certain conditional access list conditions are valid only for specific classes. For example, WHEN(SYSID) is valid only for the PROGRAM class and WHEN(CRITERIA) is valid only for general resource classes (not data sets). When copying the conditional access list from profile-name-2 to profile-name-1, the profile may differ if the condition is not valid for the data set class. For example, if profile-name-2 is a PROGRAM profile with SYSID or CRITERIA entries in the conditional access list, those entries are not copied to the new data set profile (profile-name-1).

- RACF places the user on the access list with ALTER access authority or, if the user is already on the access list, changes the user's access authority to ALTER. This does not occur if the NOADDCREATOR option is in effect.

If the profile being added is for a group data set and the user has the GRPACC attribute for that group, RACF places the group on the access list with UPDATE access authority. If the group is already on the access list, RACF changes the group's access authority to UPDATE. These access list changes do not occur if the data set profile is created only because the user has the OPERATIONS attribute.

- The security label, if specified in the model profile, is not copied. Instead, the user's current security label is used.

- Information in the non-RACF segments (for example, the DFP segment) is not copied.

FVOLUME(profile-name-2-serial)

Specifies the volume RACF is to use to locate the model profile (profile-name-2).

If you specify FVOLUME and RACF does not find profile-name-2 associated with that volume, the command fails. If you omit this operand and the data set name appears more than once in the RACF database, the command fails.

FVOLUME is valid only when FCLASS either specifies or defaults to DATASET and when profile-name-2 specifies a discrete profile. Otherwise, RACF ignores FVOLUME.
ADDSD

GENERIC
Specifies that RACF is to treat profile-name-1 as a fully qualified generic name, even if it does not contain any generic characters.

MODEL
Specifies that you are defining a model profile to be used when new data sets are created. The SETROPTS command (specifying MODEL operand with either GROUP or USER) controls whether this profile is used for data sets with group names or user ID names.

When you specify MODEL, you can omit UNIT and VOLUME.

When you specify MODEL, the SET, GENERIC, and TAPE operands are ignored, and NOSET is used as the default.

MODEL and GENERIC operands are mutually exclusive. You cannot specify a generic profile for automatic profile modelling through the MODEL operand of ADDUSER, ALTUSER, ADDGROUP, or ALTGROUP. However, you can explicitly use a generic profile as a model with the FROM operand, and if needed, the FG generic operand of the ADDSD command.

For information about automatic profile modeling, refer to z/OS Security Server RACF Security Administrator’s Guide.

TAPE
Specifies that the data set profile is to protect a tape data set. If tape data set protection is not active, RACF treats TAPE as an invalid operand and issues an appropriate error message. If profile-name-1 is a generic profile name, RACF ignores this operand. (RACF processes a tape data set protected by a generic profile in the same way as it processes a DASD data set protected by a generic profile.)

LEVEL(nn)
Specifies a level indicator, where nn is an integer from 0 and 99. The default is 0.

Your installation assigns the meaning of the value.

RACF includes it in all records that log data set accesses and in the LISTDSD command display.

SET | SETONLY | NOSET
If you do not specify SET, SETONLY, or NOSET, the default value is SET.

SET
Specifies that the data set is to be RACF-indicated. SET is the default value when you are RACF-protecting a data set. If the indicator is already on, the command fails. If you specify a generic profile name or the GENERIC operand, RACF ignores this operand.

SETONLY
Specifies that for a tape data set, RACF is to create only an entry in the TVTOC; it is not to create a discrete data set profile. Specifying SETONLY allows you to protect a tape data set with a TVTOC and a generic profile.

Thus, you would normally specify SETONLY with TAPE, and, when you do, RACF ignores the OWNER, UACC, AUDIT, DATA, WARNING, LEVEL, and RETPD operands. If you specify SETONLY without TAPE, RACF treats SETONLY as SET.

NOSET
Specifies that the data set is not to be RACF-indicated.
ADDSD

For a DASD data set, use NOSET when you are defining a data set to RACF that has been brought from another system where it was RACF-protected. (The data set is already RACF-indicated.)

For a tape data set, use NOSET when, because of a previous error, the TVTOC indicates that the data set is RACF-indicated, but the discrete profile is missing.

If you specify NOSET, for a discrete profile, when the data set is not already RACF-indicated, RACF access control to that data set is not enforced.

If you specify NOSET, the volumes on which the data set or catalog resides need not be online, and the password in the first operand of this command is not required.

To use NOSET, one of the following must be true:

- You must have the SPECIAL attribute
- The profile must fall within the scope of a group in which you have the group-SPECIAL attribute
- The high-level qualifier of the data set name (or the qualifier supplied by a command installation exit routine) must be your user ID.

If you specify a generic profile name, RACF ignores this operand.

**Note:** If you specify a profile name that exists as a generation data group (GDG) data set base name with NOSET—but do not specify a unit and volume, RACF creates a model profile for the data set instead of a discrete profile. In this situation, the model profile provides the same protection as a discrete profile.

**NOTIFY(userid)**

Specifies the user ID of a RACF-defined user to be notified whenever RACF uses this profile to deny access to a data set. If you specify NOTIFY without userid, RACF takes your user ID as the default; you are notified whenever the profile denies access to a data set.

A user who is to receive NOTIFY messages should log on frequently, both to take action in response to the unauthorized access attempts the messages describe and to clear the messages from the SYS1.BRODCAST data set. (When the profile also includes WARNING, RACF might have granted access to the data set to the user identified in the message.)

**Note:** The user ID specified on the NOTIFY operand is not notified when the profile disallows creation or deletion of a data set. NOTIFY is used only for resource access checking, not for resource creation or deletion.

**OWNER(userid or group-name)**

Specifies a RACF-defined user or group to be assigned as the owner of the data set profile. When you define a group data set, the user you designate as owner must have at least USE authority in the group specified by the high-level qualifier of the data set name (or the qualifier determined by the naming conventions routine or by a command installation exit routine).

If you omit this operand, you are defined as the owner of the data set profile. However, if the high-level qualifier is a user ID that is different from your user ID, the OWNER of the profile is the user ID specified in the high-level qualifier. In addition, if you are using naming convention processing, either through the naming convention table or an exit, the owner of the profile is determined by the naming convention processing. If you have the SPECIAL attribute and define a profile for a group data set while SETROPTS ADDCREATOR is in effect, your
user ID is added to the access list for the data set with ALTER access authority, whether or not you specify the OWNER operand. If you have the SPECIAL attribute and define a profile for a user data set, your user ID is not added to the access list for the data set.

If you specify OWNER(userid), the user you specify as the owner does not automatically have access to the data set. Use the PERMIT command to add the owner to the access list as desired. If you specify OWNER(group-name), RACF treats any users who have the group-SPECIAL attribute in the group as owners of the data set profile.

RETPD(nnnnn)
Specifies the RACF security retention period for a tape data set. The security retention period is the number of days that must elapse before a tape data set profile expires. (Note that, even though the data set profile expires, RACF-protection for data sets protected by the profile is still in effect. For more information, see \textit{z/OS Security Server RACF Security Administrator's Guide}.)

The number you specify, nnnnn must be one to five digits in the range of 0 through 65533. To indicate a data set that never expires, specify nnnnn as 99999. When 99999 is used, the SETROPTS command stores it internally as 65534.

The RACF security retention period is the same as the data set retention period specified by the EXPDT/RETPD parameters on the JCL DD statement only when the data set profile is discrete and you do not modify the RACF security retention period.

When the TAPEVOL class is active, RACF checks the RACF security retention period before it allows a data set to be overwritten. RACF adds the number of days in the retention period to the creation date for the data set. If the result is less than the current date, RACF continues to protect the data set.

When the TAPEVOL class is not active, RACF ignores the RETPD operand.

If you omit RETPD and your installation has established a default security retention period (through the RETPD operand on the SETROPTS command), RACF uses the default. If you omit RETPD and your installation has not established a default, RACF uses 0 as a default.

Specifying this operand for a DASD data set does not cause an error, but it has no meaning because RACF ignores the operand during authorization checking.

SECLABEL(security-label)
Specifies the name of an installation-defined security label representing an association between a particular security level and a set of zero or more categories.

A security label corresponds to a particular security level (such as CONFIDENTIAL) with a set of zero or more security categories (such as PAYROLL or PERSONNEL).

RACF stores the name of the security label you specify in the data set profile if you are authorized to use that label.

If you are not authorized to use the security label or if the name you had specified is not defined as a SECLABEL profile in the SECLABEL class, the data set profile is not created.

SECLEVEL(security-level)
Specifies the name of an installation-defined security level. This name
corresponds to the number that is the minimum security level a user must have to access the data set. security-level must be a member of the SECLEVEL profile in the SECDATA class.

When you specify SECLEVEL and the SECDATA class is active, RACF adds security level access checking to its other authorization checking. If global access checking does not grant access, RACF compares the security level allowed in the user profile with the security level required in the data set profile. If the security level in the user profile is less than the security level in the data set profile, RACF denies the access. If the security level in the user profile is equal to or greater than the security level in the data set profile, RACF continues with other authorization checking.

Note: RACF does not perform security level checking for a started task or user that has the RACF privileged or trusted attribute. The RACF privileged or trusted attribute can be assigned to a started task through the RACF started procedures table or STARTED class, or to other users by installation-supplied RACF exits.

If the SECDATA class is not active, RACF still stores the security-level you specified in the data set profile, but cannot perform security level checking until you have activated the SECDATA class. If the name you specify is not defined as a SECLEVEL profile and the SECDATA class is active, you are prompted to provide a valid name for security-level.

TME
Specifies that information for the Tivoli Security Management Application is to be added.

Note: The TME segment fields are intended to be updated only by the Tivoli Security Management application, which manages updates, permissions, and cross references. A security administrator should only directly update TME fields on an exception basis.

ROLES(role-access-specification ...)
Specifies a list of roles and associated access levels related to this profile.

One or more role-access-specification values can be specified, each separated by blanks. Each value should contain no imbedded blanks and should have the following format:

d:authority[:conditional-class:conditional-profile]

where role-name is a discrete general resource profile defined in the ROLE class. The authority is the access authority (NONE, EXECUTE, READ, UPDATE, CONTROL, or ALTER) with which groups in the role definition should be permitted to the resource.

The conditional-class is a class name (APPCPORT, CONSOLE, JESINPUT, PROGRAM, TERMINAL, or SYSID) for conditional access permission, and is followed by the conditional-profile value, a resource profile defined in the conditional class.

UACC(access-authority)
Specifies the universal access authority to be associated with the data sets. The universal access authorities are ALTER, CONTROL, UPDATE, READ, EXECUTE, and NONE. If you omit UACC or specify UACC with no access authority, RACF uses the default value in your current connect group. If you specify CONTROL for a tape data set or a non-VSAM DASD data set, RACF
treats the access authority as UPDATE. If you specify EXECUTE for a tape
data set, or a DASD data set not used as a program library, RACF treats the
access authority as NONE.

If a user accessing a data set has the RESTRICTED attribute, RACF treats the
universal access authority (UACC) as NONE for that access attempt.

UNIT(type)
Specifies the unit type on which a tape data set or a non-VSAM DASD data set
resides. You can specify an installation-defined unit name, a generic device
type, or a specific device address. If you specify UNIT and VOLUME for a
DASD data set, RACF assumes that the data set is a non-VSAM data set;
therefore, do not use UNIT and VOLUME for a VSAM data set.

If the data set is not cataloged, UNIT and VOLUME are required. You must
specify UNIT and VOLUME for data sets cataloged with an esoteric name (such
as an installation-defined unit name).

If you specify a generic or model profile name, RACF ignores this operand.

VOLUME(volume-serial ...)
Specifies the volumes on which a tape data set or a non-VSAM DASD data set
resides. If you specify UNIT and VOLUME for a DASD data set, RACF
assumes that the data set is a non-VSAM data set; therefore, do not use UNIT
and VOLUME for a VSAM data set.

If the data set is not cataloged, UNIT and VOLUME are required. You must
specify UNIT and VOLUME for data sets cataloged with an esoteric name (such
as an installation-defined unit name).

If you specify a tape data set profile name, you can specify only one volume.
If you specify a generic or model profile name, RACF ignores this operand.

WARNING
Specifies that even if access authority is insufficient, RACF is to issue a
warning message and allow access to the resource. RACF also records the
access attempt in the SMF record if logging is specified in the profile.

When SETROPTS MLACTIVE(FAILURES) is in effect: A user or task can
access a data set that is in WARNING mode and has no security label even
when MLACTIVE(FAILURES) is in effect and the class requires security labels.
The user or task receives a warning message and gains access.

Examples
Table 4. ADDSD examples
Example 1

Operation  User ADM1 wants to create a generic profile to protect all data sets
having the high-level qualifier SALES. Only users with a security level of
CONFIDENTIAL or higher are to be able to access the data sets.

Known  User ADM1 has the SPECIAL attribute and the installation has defined
CONFIDENTIAL as a valid security level name. User ADM1 wants to
issue the command as a RACF TSO command.

Command  ADDSD 'SALES,.*' UACC(READ) AUDIT(ALL(READ))
          SECLEVEL(CONFIDENTIAL)

Defaults  OWNER(ADM1) LEVEL(0)
### ADDSD

#### Table 4. ADDSD examples (continued)

**Example 2**

**Operation**
User AEH0 wants to protect the data set AEH0.DEPT1.DATA with a discrete RACF profile.

**Known**
User AEH0 is RACF-defined. AEH0.DEPT1.DATA is not cataloged. It resides on volume USER03 which is a 3330 volume. User AEH0 wants to issue the command as a RACF TSO command.

**Command**
```
ADDSD 'AEH0.DEPT1.DATA' UNIT(3330) VOLUME(USER03)
```

**Defaults**
- OWNER(AEH0)
- UACC(UACC of user AEH0 in current connect group)
- AUDIT(FAILURES(READ))
- LEVEL(0)
- SET

**Example 3**

**Operation**
User ADM1 wants to RACF-define the DASD data set SYS1.ICH02.DATA which was brought from another system where it was protected by a discrete RACF profile and was RACF-indicated. On the new system, only users with a security category of DEPT1 are to be allowed to access the data set.

**Known**
User ADM1 has the SPECIAL attribute. SYS1.ICH02.DATA is cataloged. User ADM1 has create authority in group SYS1 and is connected to group SYS1 with the group-SPECIAL attribute. The installation has defined DEPT1 as a valid security category. User ADM1 wants to issue the command as a RACF TSO command.

**Command**
```
ADDSD 'SYS1.ICH02.DATA' OWNER(SYS1) UACC(NONE) AUDIT(ALL) NOSET CATEGORY(DEPT1)
```

**Defaults**
- LEVEL(0)

**Example 4**

**Operation**
User AEHO wants to create a model profile for group RSC and place an installation-defined description in the profile.

**Known**
User AEHO has at least CREATE authority in group RSC. User AEHO wants to issue the command as a RACF TSO command.

**Command**
```
ADDSD 'RSC.ACCESS.PROFILE' MODEL DATA('PROFILE THAT CONTAINS MODELING INFORMATION')
```

**Defaults**
- OWNER(AEHO), UACC(the UACC of user AEHO in current group)
- AUDIT(FAILURES(READ))
- LEVEL(0)

**Example 5**

**Operation**
User AEH1 wants to protect the tape data set named AEH1.TAPE.RESULTS with a discrete RACF profile.

**Known**
User AEH1 is a RACF-defined user. Data set AEH1.TAPE.RESULTS is cataloged, and tape data set protection is active. User AEH1 wants to issue the command as a RACF TSO command.

**Command**
```
ADDSD 'AEH1.TAPE.RESULTS' UACC(NONE) AUDIT(ALL(READ)) TAPE NOTIFY FILESEQ(1) RETPD(100)
```

**Defaults**
- LEVEL(0)

**Example 6**

**Operation**
User AEH1 wants to protect the tape data set named AEH1.TAPE.FUTURES with a discrete RACF profile, which is so much like the profile created for AEH1.TAPE.RESULTS (Example 5) that AEH1 can use the existing profile as a model for the new profile.

**Known**
User AEH1 is a RACF-defined user. Data set AEH1.TAPE.FUTURES is cataloged, and tape data set protection is active. User AEH1 wants to issue the command as a RACF TSO command.

**Command**
```
ADDSD 'AEH1.TAPE.FUTURES' FROM('AEH1.TAPE.RESULTS') FILESEQ(2)
```

**Defaults**
- LEVEL(0)
Table 4. ADDSD examples (continued)

Example 7

**Operation**
User ADM1 wants to create a generic profile to protect all data sets having the high-level qualifier PROJECTA. The data sets protected by the profile will be managed by DFP. Group TEST4 will be assigned as the actual owner of the data sets protected by the profile. The profile will have a universal access authority of READ.

User ADM1 wants to direct the command to run at the local node under the authority of user DAP02 and prohibit the command from being automatically directed to other nodes.

**Known**
Users ADM1 and DAP02 have the SPECIAL attribute. TEST4 is a RACF-defined group. Users ADM1 and DAP02 have an already established user ID association. User ADM1 wants to issue the command as a RACF TSO command.

**Command**
ADDSD 'PROJECTA.*' UACC(READ) DFP(RESOWNER(TEST4)) ONLYAT(.DAP02)

**Defaults**
OWNER(ADM1) LEVEL(0) AUDIT(FAILURES(READ))

**Results**
The command is only processed on the local node and not automatically directed to any other nodes in the RRSF configuration.

Example 8

**Operation**
User TSO7 wants to create a generic profile to protect all data sets having the high-level qualifier PROJECTB with a security label of CONF. User TSO7 is authorized to the security label. User TSO7 wants to issue the command as a RACF operator command, and the RACF subsystem prefix is @.

**Known**
User TSO7 is a RACF-defined user.

**Command**
@ADDSD 'PROJECTB.*' SECLABEL(CONF)

**Defaults**
None.
ADDUSER

ADDUSER (Add user profile)

Purpose

Use the ADDUSER command to define a new user to RACF and establish the user’s relationship to an existing RACF-defined group.

The command adds a profile for the new user to the RACF database and creates a connect profile that connects the user to whichever default group you specify.

The user profile consists of a RACF segment and, optionally, other segments such as a TSO segment, a DFP segment, or an OMVS segment. You can use this command to define information in any segment of the user’s profile.

Although user ID association information is in the user’s profile, you must use the RACLINK command to define a user ID association.

Attention:

- When the ADDUSER command is issued from ISPF, the TSO command buffer (including password and password phrase data) is written to the ISPLOG data set. As a result, you should not issue this command from ISPF or you must control the ISPLOG data set carefully.

- If the ADDUSER command is issued as a RACF operator command, the command and all data (including password and password phrase data) is written to the system log. You should not issue the ADDUSER command as an operator command unless specifying NOPASSWORD. For all other cases you should execute it as a TSO command.

This command is not intended to be used for profiles in the DIGTCERT or DIGTNMAP classes.

Issuing options

The following table identifies the eligible options for issuing the ADDUSER command:

<table>
<thead>
<tr>
<th>As a RACF TSO command?</th>
<th>As a RACF operator command?</th>
<th>With command direction?</th>
<th>With automatic command direction?</th>
<th>From the RACF parameter library?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

For information on issuing this command as a RACF TSO command, refer to Chapter 3, “RACF TSO commands,” on page 15.

For information on issuing this command as a RACF operator command, refer to Chapter 4, “RACF operator commands,” on page 21.

You must be logged on to the console to issue this command as a RACF operator command.

Related commands

- To change a user profile, see ALTUSER (Alter user profile)
- To delete a user profile, see DELUSER (Delete user profile)
- To list a user profile, see LISTUSER (List user profile)
To administer user ID associations, see RACLINK (Administer user ID associations).

To obtain a list of user profiles, see SEARCH (Search RACF database).

Authorization required

When issuing the ADDUSER command as a RACF operator command, you might require sufficient authority to the proper resource in the OPERCMDS class. See z/OS Security Server RACF Security Administrator’s Guide for further information.

To use the ADDUSER command, you must have one of the following:

- The SPECIAL attribute
- The CLAUTH attribute for the USER class while one of the following is true:
  - You are the owner of the default group specified in this command.
  - You have JOIN authority in the default group specified in this command.
  - The default group is within the scope of a group in which you have the group-SPECIAL attribute.

You must have the SPECIAL attribute to give the new user the OPERATIONS, SPECIAL, or AUDITOR attribute. You need not have the SPECIAL attribute to specify the OWNER operand.

You cannot assign a user an attribute or authority higher than your own.

To assign a security category to a profile, you must have the SPECIAL attribute, or the category must be in your user profile. To assign a security level to a profile, you must have the SPECIAL attribute or, in your own profile, a security level that is equal to or greater than the security level you are assigning.

To assign a security label when SETROPTS SECLABELCONTROL is active, you must have the SPECIAL attribute.

To define information within a segment other than the base segment, you must have one of the following:

- The SPECIAL attribute
- At least UPDATE authority to the desired field within the segment through field-level access control.

For information on field-level access checking, see z/OS Security Server RACF Security Administrator’s Guide.

To specify the AT keyword, you must have READ authority to the DIRECT.node resource in the RRSFDATA class and a user ID association must be established between the specified node.userid pair(s).

To specify the ONLYAT keyword you must have the SPECIAL attribute, the userid specified on the ONLYAT keyword must have the SPECIAL attribute, and a user ID association must be established between the specified node.userid pair(s) if the user IDs are not identical.

To specify the SHARED keyword, you must have the SPECIAL attribute or at least READ authority to the SHARED.IDS resource in the UNIXPRIV class.
ADDUSER

Syntax

For the key to the symbols used in the command syntax diagrams, see "Syntax of RACF commands and operands" on page 9. The complete syntax of the ADDUSER command is:

```
[subsystem-prefix][ADDUSER | AU]
  
  (userid ...)
  
  [ ADDCATEGORY(category-name ...) ]
  [ ADSP | NOADSP ]
  [ AT([node].userid ... | ONLYAT([node].userid ...) ]
  [ AUDITOR | NOAUDITOR ]
  [ AUTHORITY(group-authority) ]
  [ CICS( 
     [ OPCLASS(operator-class ...) ]
     [ OPIDENT(operator-id) ]
     [ OPPRTY(operator-priority) ]
     [ RSLKEY(rslkey ... | 0 | 1 | 99 ) ]
     [ TIMEOUT(timeout-value) ]
     [ TSLKEY(tslkey ... | 0 | 1 | 99 ) ]
     [ XRFSOFF( FORCE | NOFORCE ) ]
  ) ]
  [ CLAUTH(class-name ...) | NOCLAUTH ]
  [ CSDATA(
     [ custom-field-name(custom-field-value) ] ...
  ) ]
  [ DATA('installation-defined-data') ]
  [ DCE( 
     [ AUTOLOGIN(YES | NO) ]
     [ DCENAME(user-principal-name) ]
     [ HOMECELL(dce-cell-name) ]
     [ HOMEUUID(home-cell-UUID) ]
     [ UUID(universal-unique-identifier) ]
  ) ]
  [ DFLTGRP(group-name) ]
  [ DFP( 
     [ DATAAPPL(application-name) ]
     [ DATACLAS(data-class-name) ]
     [ MGMTCLAS(management-class-name) ]
     [ STORCLAS(storage-class-name) ]
  ) ]
  [ EIM( 
     [ LDAPPROF(ldapbind_profile)
     ]
  ) ]
  [ GRPACC | NOGRPACC ]
  [ KERB( 
     [ ENCRYPT ( 
       [ DES | NODES ]
       [ DES3 | NODES3 ]
       [ DESD | NODESD ]
       [ AES128 | NOAES128 ]
       [ AES256 | NOAES256 ]
     ) ]
     [ KERBNAME(kerberos-principal-name) ]
     [ MAXTKTLFE(max-ticket-life) ]
  ) ]
```
ADDUSER

```
[ LANGUAGE(
    [ PRIMARY(language) ]
    [ SECONDARY(language) ]
  )
]

[ LNOTES(
    [ SNAME(short-name) ]
  )
]

[ MODEL(dname) ]
[ NAME(user-name) ]
[ NDS(
    [ UNAME(user-name) ]
  )
]

[ NETVIEW(
    [ CONSNAME(console-name) ]
    [ CTL(GENERAL | GLOBAL | SPECIFIC) ]
    [ DOMAINS(domain-name ... ) ]
    [ IC('command | command-list') ]
    [ MSGRECVR(NO | YES ) ]
    [ NGMADMNN(NO | YES ) ]
    [ NGMFVSPN(view-span) ]
    [ OPCLASS(class ... ) ]
  )
]

[ OIDCARD | NOOIDCARD ]

[ OMVS((
    [ ASSIZEMAX(address-space-size) ]
    [ AUTOID | UID(user-identifier) ]
    [ CPUTIMEMAX(cpu-time) ]
    [ FILEPROCMAX(files-per-process) ]
    [ HOME(initial-directory-name) ]
    [ MEMLIMIT(nonsaged-memory-size) ]
    [ MMAPAREAMAX(memory-map-size) ]
    [ PROCUSERMAX(processes-per-UID) ]
    [ PROGRAM(program-name) ]
    [ SHMEMMAX(shared-memory-size) ]
    [ THREADS(MAX(threads-per-process) ]
  ))
]

[ OPERATIONS | NOOPERATIONS ]

[ OPERPARM(
    [ ALTGRP(alternate-console-group) ]
    [ AUTH(operator-authority) ]
    [ AUTO( YES | NO ) ]
    [ CMDSYS(system-name) ]
    [ DOM( NORMAL | ALL | NONE ) ]
    [ HC( YES | NO ) ]
    [ INTIDS( YES | NO ) ]
    [ KEY(searching-key) ]
    [ LEVEL(message-level) ]
    [ LOGCMDRESP( SYSTEM | NO ) ]
    [ MFORM(message-format) ]
    [ MIGID( YES | NO ) ]
    [ MONITOR(event) ]
    [ MSCOPE( system-names | * | ALL ) ]
    [ ROUTCODE( ALL | NONE | routing-codes ) ]
    [ STORAGE(amount) ]
    [ UD( YES | NO ) ]
    [ UNKNIDS( YES | NO ) ]
  )
]```
ADDUSER

[ OVM( [ FSROOT(file-system-root) ] [ HOME(initial-directory-name) ] [ PROGRAM(program-name) ] [ UID(user-identifier) ] ) ]
[ OWNER(userid or group-name) ]
[ PASSWORD(password) | NOPASSWORD ]
[ PHRASE("password-phrase") ]
[ PROXY([ [ LDAPHOST(ldap_url) ] [ BINDDN(bind_distinguished_name) ] [ BINDPW(bind_password) ] ] ) ]
[ RESTRICTED | NORESTRICTED ]
[ SECLABEL(seclabel-name) ]
[ SECLEVEL(secellevel-name) ]
[ SPECIAL | NOSPECIAL ]
[ TSO( [ ACCTNUM(account-number) ] [ COMMAND(command-issued-at-logon) ] [ DEST(destination-id) ] [ HOLDCLASS(hold-class) ] [ JOBCLASS(job-class) ] [ MAXSIZE(maximum-region-size) ] [ MSGCLASS(message-class) ] [ PROC(logon-procedure-name) ] [ SECLABEL(security-label) ] [ SIZE(default-region-size) ] [ SYS(sysout-class) ] [ UNIT(unit-name) ] [ USERDATA(user-data) ] ) ]
[ UACC(access-authority) ]
[ WHEN( [ DAYS(day-info) ] [ TIME(time-info) ] ) ]
[ WORKATTR( [ WAACNT(account-number) ] [ WAADDR1(address-line-1) ] [ WAADDR2(address-line-2) ] [ WAADDR3(address-line-3) ] [ WAADDR4(address-line-4) ] [ WABLDG(building) ] [ WADEPT(department) ] [ WANAME(name) ] [ WARoom(room) ] ) ]

For information on issuing this command as a RACF TSO command, refer to Chapter 3, “RACF TSO commands,” on page 15.

For information on issuing this command as a RACF operator command, refer to Chapter 4, “RACF operator commands,” on page 21.
Parameters

**subsystem-prefix**

Specifies that the RACF subsystem is the processing environment of the command. The subsystem prefix can be either the installation-defined prefix for RACF (1–8 characters) or, if no prefix has been defined, the RACF subsystem name followed by a blank. If the command prefix was registered with CPF, you can use the MVS command D OPDATA to display it or you can contact your RACF security administrator.

Only specify the subsystem prefix when issuing this command as a RACF operator command. The subsystem prefix is required when issuing RACF operator commands.

**userid**

Specifies the user to be defined to RACF. If you are defining more than one user, the list of user IDs must be enclosed in parentheses.

This operand is required and must be the first operand following ADDUSER.

Each user ID must be unique and must not currently exist on the RACF database as a user ID or a group name.

**ADDCATEGORY(category-name ...)**

Specifies one or more names of installation-defined security categories. The names you specify must be defined as members of the CATEGORY profile in a SECDATA class. (For information on defining security categories, see [z/OS Security Server RACF Security Administrator's Guide](#).)

When the SECDATA class is active and you specify ADDCATEGORY, RACF performs security category checking in addition to its other authorization checking. If a user requests access to a resource, RACF compares the list of security categories in the user’s profile with the list of security categories in the resource profile. If RACF finds any security category in the resource profile that is not in the user’s profile, RACF denies access to the resource. If the user’s profile contains all the required security categories, RACF continues with other authorization checking.

**Note:** RACF does not perform security category checking for a started task or user with the RACF privileged or trusted attribute. The RACF privileged or trusted attribute can be assigned to a started task through the RACF started procedures table or STARTED class, or to other users by installation-supplied RACF exits.

**ADSP | NOADSP**

**ADSP**

Specifies that all permanent tape and DASD data sets created by the new user are to be automatically RACF-protected by discrete profiles. ADSP specified on the ADDUSER command overrides NOADSP specified on the CONNECT command.

If SETROPTS NOADSP is in effect, RACF ignores the ADSP attribute at logon or job initiation.

**NOADSP**

Specifies that the new user is not to have the ADSP attribute. NOADSP is the default value if you omit both ADSP and NOADSP.

**AT | ONLYAT**

The AT and ONLYAT keywords are only valid when the command is issued as a RACF TSO command.
ADDUSER

AT((node).userid ...)  
Specifies that the command is to be directed to the node specified by node, 
where it runs under the authority of the user specified by userid in the 
RACF subsystem address space.

If node is not specified, the command is directed to the local node.

ONLYAT((node).userid ...)  
Specifies that the command is to be directed only to the node specified by 
ode where it runs under the authority of the user specified by userid in the 
RACF subsystem address space.

If node is not specified, the command is directed only to the local node.

AUDITOR | NOAUDITOR

AUDITOR  
Specifies that the new user has full responsibility for auditing the use of 
system resources, and is able to control the logging of detected accesses 
to any RACF-protected resources during RACF authorization checking and 
accesses to the RACF database.

You must have the SPECIAL attribute to enter the AUDITOR operand.

NOAUDITOR  
Specifies that the new user does not have the AUDITOR attribute. 
NOAUDITOR is the default value if you omit both AUDITOR and 
NOAUDITOR.

AUTHORITY(group-authority)  
Specifies the level of group authority for the new user in the default group. The 
valid group authority values are USE, CREATE, CONNECT, and JOIN, as 
described in "Group authorities" on page 13. If you omit this operand or specify 
AUTHORITY without group-authority, the default value is USE.

This operand is group-related. If a user is connected to other groups (with the 
CONNECT command), the user can have a different group authority in each 
group.

CICS  
Defines CICS operator information for a new CICS terminal user. You can 
control access to an entire CICS segment or to individual fields within the CICS 
segment by using field-level access checking. For more information, see Z/OS 

OPCLASS(operator-class ...)  
Specifies numbers 1–24, defined as two digits, representing classes 
assigned to this operator to which BMS (basic mapping support) messages 
are to be routed.

OPIDENT(operator-id)  
Specifies a 1–3 character identification of the operator for use by BMS.

Operator identifiers can consist of any characters, and can be entered with 
or without single quotation marks. The following rules apply:

• If parentheses, commas, blanks, or semicolons are to be entered as part 
of the operator identifier, the character string must be enclosed in single 
quotation marks. For example, if the operator identifier is (1), you must 
enter OPIDENT(’(1)’).
If a single quotation mark is intended to be part of the character string, use two single quotation marks together for each single quotation mark within the string, and enclose the entire string within single quotation marks.

If OPIDENT is not specified, the field defaults to blanks in the RACF user profile, and blanks appear in the field in the LISTUSER command output.

**OPPRTY**(*operator-priority*)
Specifies the number from 0–255 that represents the priority of the operator.

If OPPRTY is not specified, the field defaults to zeros in the RACF user profile, and zeros appear in the field in the LISTUSER command output.

**RSLKEY**(*rslkey ... | 0 | 99*)
Specifies the resource security level (RSL) keys assigned to the user. The RSL keys are used by CICS on distributed platforms. Each CICS resource has one RSL key assigned to it; in order for a user to access a resource, the user must have the same RSL key as the RSL key assigned to the resource.

- **RSLKEY**(*rslkey ...*) specifies a list of one or more numbers in the range of 1 through 24 which represent the resource security level (RSL) keys assigned to the user.
- If RSLKEY(0) is specified, no RSL keys are assigned to the user.
- If RSLKEY(99) is specified, all RSL keys are assigned to the user (1–24, inclusive).
- Keys 0 and 99 are mutually exclusive and cannot be specified with any other keys.
- If RSLKEY is specified with no key numbers, RSLKEY(0) is defaulted.
- If RSLKEY is not specified, CICS will treat it as RSLKEY(0).

**TIMEOUT**(*timeout-value*)
Specifies the time, in hours and minutes, that the operator is allowed to be idle before being signed off. The value for TIMEOUT can be entered in the form *m*, *mm*, *hmm*, *hhmm*, where the value for *m* or *mm* must be 00–59, or 00–60 when *h* or *hh* is not specified or is specified as 0 or 00. The value for *h* or *hh* must be 00–99.

TIMEOUT defaults to 0 if omitted, meaning no timeout.

**TSLKEY**(*tslkey ... | 0 | 1 | 99*)
Specifies the transaction security level (TSL) keys assigned to the user. The TSL keys are used by CICS on distributed platforms. Each CICS transaction has one TSL key assigned to it; in order for a user to run a transaction, the user must have the same TSL key as the TSL key assigned to the transaction.

- **TSLKEY**(*tslkey ...*) specifies a list of one or more numbers in the range of 1 through 64 which represent the transaction security level (TSL) keys assigned to the user.
- If TSLKEY(0) is specified, no TSL keys are assigned to the user.
- If TSLKEY(99) is specified, all TSL keys are assigned to the user (1–64, inclusive).
- Keys 0 and 99 are mutually exclusive and cannot be specified with any other keys.
- If TSLKEY is specified with no key numbers, TSLKEY(1) is defaulted.
ADDUSER

- If TSLKEY is not specified, CICS will treat it as TSLKEY(1).

**XRFSOFF(FORCE | NOFORCE)**

FORCE means that the user is signed off by CICS when an XRF takeover occurs.

**CLAUTH | NOCLAUTH**

**CLAUTH(class-name ...)**

Specifies the classes in which the new user is allowed to define profiles to RACF for protection. Classes you can specify are USER, and any resource classes defined in the class descriptor table.

To enter the CLAUTH operand, you must have the SPECIAL attribute or have the CLAUTH attribute for the classes specified. If you do not have sufficient authority for a specified class, RACF ignores the CLAUTH specification for that class and continues processing with the next class name specified.

**Note:** The CLAUTH attribute has no meaning for the FILE and DIRECTORY classes.

**NOCLAUTH**

Specifies that the new user is not to have the CLAUTH attribute.

NOCLAUTH is the default if you omit both CLAUTH and NOCLAUTH.

**CSDATA**

Specifies information to add a custom field for this user.

Usage for each custom field is defined using the CFDEF operand of the RDEFINE command for resource profiles in the CFIELD class. Contact your security administrator to see how custom fields are used at your installation. For more information about custom fields, see z/OS Security Server RACF Administrator’s Guide.

**custom-field-name(custom-field-value) ...**

Specifies the name and value of a custom field for this user. You can add values for multiple custom fields with a single ADDUSER command.

**Rules:**

- You must use the same custom-field-name as defined by the CFIELD profile named USER.CSDATA.custom-field-name. (The CFIELD profile is defined using the CFDEF operand of the RDEFINE command.)
- You must specify a custom-field-value that is valid for the attributes of this custom field. (The attributes, such as data type, are defined in the CFDEF segment of the CFIELD profile.)

**DATA('installation-defined-data')**

Specifies up to 255 characters of installation-defined data to be stored in the user’s profile and must be enclosed in single quotation marks. It can also contain double-byte character set (DBCS) data. Note that only 254 characters are chained off the ACEE.

Use the LISTUSER command to list this information.

**DCE**

Specifies that, when you define an z/OS DCE user to RACF, you can enter any of the following suboperands to specify information for that user. Each suboperand defines information that RACF stores in a field within the DCE segment of the user’s profile.
ADDUSER

You can control access to an entire DCE segment or to individual fields within the DCE segment by using field level access checking.

To define information within the DCE segment, you must have one of the following:

- The SPECIAL attribute
- At least UPDATE authority to the desired field within the segment through field-level access control.

For information on field-level access checking, see [z/OS Security Server RACF Security Administrator's Guide](#).

---

**Important**

The ability to associate a RACF and DCE identity depends on replicated information between DCE and RACF. Do not change the user’s UUID, principal name, or cell name in either RACF or the DCE registry without a corresponding update in the other registry.

---

**AUTOLOGIN(NO | YES)**

Specifies whether z/OS UNIX DCE is to log this user into z/OS UNIX DCE automatically. If AUTOLOGIN(NO) is specified, z/OS UNIX DCE does not attempt to login this user to z/OS UNIX DCE automatically. If AUTOLOGIN is not specified, AUTOLOGIN(NO) is the default.

**DCENAME(user-principal-name)**

Specifies the DCE principal name defined for this RACF user in the DCE registry.

The DCENAME you define to RACF can contain 1–1023 characters and can consist of any character. You can enter the name with or without single quotation marks, depending on the following:

- If parentheses, commas, blanks, or semicolons are entered as part of the name, the character string must be enclosed in single quotation marks.
- If a single quotation mark is intended to be part of the name, use two single quotation marks together for each single quotation mark within the string, and enclose the entire string within single quotation marks.

Both uppercase and lowercase characters are accepted and maintained in the case in which they are entered. RACF does not ensure that a valid DCENAME has been specified.

The DCENAME assigned to a user must be the same as the DCE principal name defined to the DCE registry.

If DCENAME is not specified, the user cannot login as an z/OS UNIX DCE user automatically, even when AUTOLOGIN(YES) is specified.

**Note:** RACF does not enforce the uniqueness of each DCENAME. The DCENAME specified must match the user’s DCE principal name that is defined to the DCE registry. If the DCENAME entered does not correspond to the DCE principal name entered in the DCE registry for this user, z/OS UNIX DCE cannot correctly associate the identity of the DCE principal with the correct RACF user ID.

**HOMECELL(dce-cell-name)**

Specifies the DCE cell name defined for this RACF user.
The HOMECELL you define to RACF can contain 1–1023 characters and can consist of any character. You can enter the name with or without single quotation marks, depending on the following:

- If parentheses, commas, blanks, or semicolons are entered as part of the cell name, the character string must be enclosed in single quotation marks.
- If a single quotation mark is intended to be part of the cell name, use two single quotation marks together for each single quotation mark within the string, and enclose the entire string within single quotation marks.

Both uppercase and lowercase characters are accepted and maintained in the case in which they are entered. The fully qualified pathname should be specified. RACF does not ensure that a valid DCE cell name has been specified.

The HOMECELL assigned to a user must be the same as the DCE cell name that this user has been defined to.

If the HOMECELL is not specified, z/OS UNIX DCE single signon to DCE support assumes that the HOMECELL for this user is the same cell that this MVS system is defined to.

RACF checks that the prefix of the HOMECELL name entered has a prefix of either /.../ or /.:/.

The notation /.../ indicates that the HOMECELL name is a global domain name service (DNS) cell name or X.500 global name.

The notation /.:/ indicates that the HOMECELL name is a cell relative CDS (cell directory service) name. When determining the naming conventions used within your DCE cell, you should contact your DCE cell administrator.

HOMEUUID(home-cell-UUID)

Specifies the DCE universal unique identifier (UUID) for the cell that this user is defined to. The UUID is a 36-character string that consists of numeric and hexadecimal characters. This string must have the delimiter character (-) in positions 9, 14, 19, and 24. The general format for the UUID string is xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx, in which x represents a valid numeric or hexadecimal character.

Be careful when assigning UUIDs. The UUID cannot be randomly assigned. The HOMEUUID is the DCE UUID of the cell that this RACF user is defined to. If HOMEUUID is not specified, the LISTUSER command displays NONE for the HOMEUUID field.

Note: The HOMEUUID specified must match the UUID of the DCE cell to which this principal (specified by the DCENAME operand) is defined.

UUID(universal-unique-identifier)

Specifies the DCE universal unique identifier (UUID) of the DCE principal defined in DCENAME. The UUID is a 36-character string that consists of numeric and hexadecimal characters. This string must have the delimiter character (-) in positions 9, 14, 19, and 24. The general format for the UUID string is xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx, in which x represents a valid numeric or hexadecimal character.

Be careful when assigning UUIDs. The UUID cannot be randomly assigned. Note that RACF does not enforce the uniqueness of each UUID entered. The DCE UUID assigned to a user must be the same as the DCE UUID...
assigned when defining this RACF user to the DCE registry as a DCE principal that is being cross-linked with this RACF user ID. This DCE principal is specified using the DCENAME operand.

**DFTGRP(group-name)**

Specifies the name of a RACF-defined group to be used as the default group for the user. If you do not specify a group, RACF uses your current connect group as the default.

**Note:** You do not have to issue the CONNECT command to connect new users to their default groups.

**DFP**

Specifies that when you define a user to RACF, you can enter any of the following suboperands to specify default values for DFP data application identifier, data class, management class, and storage class. DFP uses this information to determine data management and DASD storage characteristics when a user creates a new data set.

You can control access to an entire DFP segment or to individual fields within the DFP segment by using field-level access checking. For more information, see [z/OS Security Server RACF Security Administrator's Guide](https://可观测的资源).  

**DATAAPPL(application-name)**

Specifies an 8-character DFP data application identifier.

**DATACLAS(data-class-name)**

Specifies the default data class. The maximum length of data-class-name is 8 characters.

A data class can specify some or all of the physical data set attributes associated with a new data set. During new data set allocation, data management uses the value you specify as a default unless it is preempted by a higher priority default, or overridden in some other way, for example by JCL.

**Note:** The value you specify must be a valid data class name defined for use on your system. For more information, see [z/OS Security Server RACF Security Administrator's Guide](https://可观测的资源).  

For information on defining DFP data classes, see [z/OS DFSMSdfp Storage Administration](https://可观测的资源).

**MGMTCLAS(management-class-name)**

Specifies the default management class. The maximum length of management-class-name is 8 characters.

A management class contains a collection of management policies that apply to data sets. Data management uses the value you specify as a default unless it is preempted by a higher priority default, or overridden in some other way, for example by JCL.

**Note:** The value you specify must be protected by a profile in the MGMTCLAS general resource class, and the user must be granted at least READ access to the profile. Otherwise, RACF does not allow the user access to the specified MGMTCLAS. For more information, see [z/OS Security Server RACF Security Administrator's Guide](https://可观测的资源).  

For information on defining DFP management classes, see [z/OS DFSMSdfp Storage Administration](https://可观测的资源).
ADDUSER

STORCLAS(storage-class-name)
   Specifies the default storage class. The maximum length of
   storage-class-name is 8 characters.
   A storage class specifies the service level (performance and availability) for
   data sets managed by the storage management subsystem (SMS). During
   new data set allocation, data management uses the value you specify as a
   default unless it is preempted by a higher priority default, or overridden in
   some other way (for example, by JCL).

   Note: The value you specify must be protected by a profile in the
   STORCLAS general resource class, and the user must be granted at
   least READ access to the profile. Otherwise, RACF does not allow
   the user access to the specified STORCLAS. For more information,
   see z/OS Security Server RACF Security Administrator’s Guide
   For information on defining DFP storage classes, see z/OS DFSMSdftp
   Storage Administration

EIM
   Specifies the bind information required to establish a connection with the EIM
domain.

LDAPPROF(ldapbind_profile)
   Specifies the name of a profile in the LDAPBIND class. The profile in the
   LDAPBIND class contains the name of an EIM domain and the bind
   information required to establish a connection with the EIM domain. The
   EIM services attempt to retrieve this information when it is not explicitly
   supplied through invocation parameters. Applications or other services that
   use the EIM services may instruct their callers to define a profile in the
   LDAPBIND class or the IRR.PROXY.DEFAULTS profile in the FACILITY
   class.
   The ldapbind_profile specifies the name of a profile in the LDAPBIND class
   containing the EIM domain and the LDAP bind information. The
   ldapbind_profile name may be 1–246 characters long. It is not a
   case-sensitive name.

GRPACC | NOGRPACC

GRPACC
   Specifies that any group data sets protected by DATASET profiles defined
   by the new user are automatically accessible to other users in the group.
   The group whose name is used as the high-level qualifier of the data set
   name (or the qualifier supplied by a command installation exit) has UPDATE
   access authority in the new profile. GRPACC specified on the ADDUSER
   command overrides NOGRPACC specified on the CONNECT command.

NOGRPACC
   Specifies that the new user does not have the GRPACC attribute.
   NOGRPACC is the default value if you omit both GRPACC and
   NOGRPACC.

KERB
   Specifies z/OS Integrated Security Services Network Authentication Service
   information for a user you are defining to RACF. Each subkeyword defines
   information that RACF stores in a field within the KERB segment of the user’s
   profile.
**ADDUSER**

**Note:** The RACF user password must be changed to be non-expired in order to complete the definition of the z/OS Network Authentication Service principal. The user cannot use any z/OS Network Authentication Service function until the definition is complete.

**ENCRYPT**

Specifies which keys the user (the z/OS Network Authentication Service principal) is allowed to use.

ENCRYPT is the default value when you specify KERB. The default values for ENCRYPT are DES, DES3, DESD, AES128, and AES256.

**DES | NODES**

Whether DES encrypted keys can be used.

**DES3 | NODES3**

Whether DES3 encrypted keys can be used.

**DESD | NODESD**

Whether DESD encrypted keys can be used.

**AES128 | NOAES128**

Whether AES128 encrypted keys can be used.

**AES256 | NOAES256**

Whether AES256 encrypted keys can be used.

When a principal's password changes, a key of each type is generated and stored in the principal's user profile. The use of each key is based on the z/OS Network Authentication Service configuration.

**Important:** The principal's password must be changed to ensure that a key of each type is generated and stored in the principal's user profile.

See [z/OS Integrated Security Services Network Authentication Service Administration](https://www.ibm.com) for information about how z/OS Network Authentication Service uses keys and how to customize environment variables related to keys.

**KERNAME**(*kerberos-principal-name*)

Specifies the z/OS user ID's local *kerberos-principal-name*.

The value specified for the local *kerberos-principal-name* must be unique. Consequently, a list of users cannot be specified on an ADDUSER command with the KERNAME keyword.

The *kerberos-principal-name* you define to RACF can consist of any character except the θ (X'7C') character. You can enter the name with or without single quotation marks, depending on the following:

- If parentheses, commas, blanks, or semicolons are entered as part of the name, the name must be enclosed in single quotation marks.
- If a single quotation mark is intended to be part of the name and the entire character string is enclosed in single quotation marks, you must use two single quotation marks together to represent each single quotation mark within the string.
- If the first character of the name is a single quotation mark, you must enter the string within single quotation marks, with two single quotation marks entered for that single quotation mark.

**Guideline:** Avoid using EBCDIC variant characters to prevent problems with different code pages.
ADDUSER

Both uppercase and lowercase characters are accepted and maintained in the case in which they are entered. However, RACF does not ensure that a valid kerberos-principal-name has been specified.

A local kerberos-principal-name must not be qualified with a realm name when specified with the KERBNAME keyword. However, RACF verifies that the local principal name, when fully qualified with the name of the local realm:

/.../local_realm_name/principal_name

does not exceed 240 characters. For example,

* If the local realm name is
  x
  fully qualified local principal names are prefixed with
  /.../x/
  and are limited to a maximum of 233 characters.

* If the local realm name is
  KERB390.ENDICOTT.IBM.COM
  fully qualified local principal names will be prefixed with
  /.../KERB390.ENDICOTT.IBM.COM/
  and be limited to a maximum of 210 characters.

This length verification requires that the REALM profile for the local realm KERBDFLT be defined and contain the name of the local realm, prior to the specification of local z/OS Network Authentication Service principals. Otherwise, z/OS Network Authentication Service users might not be properly defined.

Note: Because of the relationship between local realm names and local kerberos-principal-names, in which the length of a fully qualified name cannot exceed 240 characters, caution and planning must go into renaming the local realm since the combined length is only checked by RACF when a local kerberos-principal-name is added or altered. Renaming the realm should be avoided as a result.

MAXTHTLFE(max-ticket-life)

Specifies the max-ticket-life in seconds, and is represented by a numeric value from 1–2 147 483 647. Note that 0 is not a valid value.

If MAXTHTLFE is specified on the definition of a local z/OS Network Authentication Service principal, the z/OS Integrated Security Services Network Authentication Service takes the most restrictive of the value defined for the local principal and the value specified on the definition of the local realm (the KERBDFLT profile in the REALM class). Consequently, if the realm max-ticket-life is 24 hours, a principal cannot get a ticket with a longer lifetime even if the max-ticket-life is set to 48 hours in the user profile. If this field is not specified for a local principal, or if NOMAXTHTLFE has been specified, the maximum lifetime for tickets created for this principal is determined from the definition of the local z/OS Network Authentication Service realm.

LANGUAGE

Specifies the user’s preferred national languages. Specify this operand if the
ADDUSER

user is to have languages other than the system-wide defaults (established by the LANGUAGE operand on the SETROPTS command).

- If this profile is for a TSO/E user who is to establish an extended MCS console session, the languages you specify should be one of the languages specified on the LANGUAGE LANGCODE statements in the MMSLSTxx PARMLIB member. See your MVS system programmer for this information.
  For more information on TSO/E national language support, see z/OS TSO/E Customization.
- If this profile is for a CICS user, see your CICS administrator for the languages supported by CICS on your system.
  For more information, see CICS RACF Security Guide.

**PRIMARY(language)**

Specifies the user’s primary language.

**SECONDARY(language)**

Specifies the user’s secondary language.

**Notes:**

1. For the primary and secondary languages, specify either the installation-defined name of a currently active language (a maximum of 24 characters) or one of the language codes (three characters in length) for a language installed on your system.

2. The language name can be a quoted or unquoted string.

3. The same language can be specified with both PRIMARY and SECONDARY parameters.

4. If the MVS message service is not active, the PRIMARY and SECONDARY values must be a 3-character language code.

**LNOTES**

Specifies the Lotus Notes for z/OS information for the user profile being added.

**SNAME(short-name)**

Specifies the Lotus Notes for z/OS short-name of the user being defined. This value should match the name stored in the Lotus® Notes® for z/OS address book for this user, but this is not verified by the command.

The short-name you define to RACF can contain 1–64 characters. The short-name can contain the following characters: uppercase and lowercase alphabetic characters (A–Z, a–z), 0–9, & (X'50'), - (X'60'), . (X'4B'), & (X'6D'), and blanks (X'40').

If the short-name you specify contains any blanks, it must be enclosed in single quotation marks. The short-name is stripped of leading and trailing blanks.

The value specified for the short-name must be unique. Consequently, a list of users cannot be specified on an ADDUSER command with the SNAME keyword.

**MODEL(dsname)**

Specifies the name of a discrete data set profile that is used as a model when new data set profiles are created that have userid as the high-level qualifier. For this operand to be effective, the MODEL(USER) option (specified on the SETROPTS command) must be active.

RACF always prefixes the data set name with userid when it accesses the model. For information about automatic profile modeling, refer to z/OS Security Server RACF Security Administrator’s Guide.
ADDUSER

NAME(user-name)
Specifies the user name to be associated with the new user ID. You can use a maximum of 20 alphanumeric or non-alphanumeric characters. If the name you specify contains any blanks, it must be enclosed in single quotation marks.

Names longer than 20 characters are truncated to 20 characters when you enclose the name in quotation marks. However, when you specify a name longer than 20 characters without enclosing the name in quotation marks, you receive an error from the TSO parse routine.

If you omit the NAME operand, RACF uses a default of twenty # (X'7B') characters (###...). Note, however, that the corresponding entry in a LISTUSER output is the word UNKNOWN.

NDS
Specifies the Novell Directory Services for OS/390 information for the user profile being added.

UNAME(user-name)
Specifies the Novell Directory Services for OS/390 user-name of the user being defined. The user-name value should match the name stored in the Novell Directory Services for OS/390 directory for this user, but this is not verified by the command.

The user-name you define to RACF can contain 1–246 characters. However, the user-name cannot contain the following characters: * (X'5C'), + (X'4E'), | (X'4F'), = (X'7E'), : (X'7A'), ; (X'5E'), ¢ (X'4A'), and brackets [ and ] (X'AD' and X'BD').

If the user-name you specify contains any parentheses or blanks, it must be enclosed in single quotation marks. The user-name is stripped of leading and trailing blanks. If a single quotation mark is intended to be part of the user-name, use two single quotation marks together for each single quotation mark within the string, and enclose the entire string within single quotation marks.

The value specified for the user-name must be unique. Consequently, a list of users cannot be specified on an ADDUSER command with the UNAME keyword.

NETVIEW

CONSNAME(console-name)
Specifies the default master console station (MCS) console name used for this operator. This default console name is used when the operator does not specify a console name on the NetView GETCONID command.

The console-name value is an identifier 1–8 characters in length whose validity is checked by MVS processing when the operator tries to use it. See z/OS MVS Planning: Operations for information on valid values for a particular release.

CTL
Specifies whether a security check is performed for this NetView operator when they try to use a span or try to do a cross-domain logon.

GENERAL
Specifies that checking is done as described for SPECIFIC, and, in addition, that the operator is allowed to access devices that are not part of any span.
GLOBAL
   Specifies that no checking is done.

SPECIFIC
   Specifies that the operator is allowed to control only devices that are in
   spans the operator started, and that a security check is to be performed
   through RACROUTE REQUEST=AUTH whenever this operator
   attempts to use a span. Also, any cross-domain logon must be to a
   domain listed in the operator’s NETVIEW segment with the DOMAINS
   keyword.

   SPECIFIC is the default.

DOMAINS(domain-name ...)
   Specifies the identifiers of NetView programs in another NetView domain
   where this operator can start a cross-domain session. The NetView
   program identifiers are coded on the NCCFID definition statement for the
   other domains, and represent the name given to that NetView program on
   the APPL statement.

   Domain-name is a 1–5 character identifier. The characters can be
   alphabetic, numeric, or national.

IC('command | command-list')
   Specifies the command or command list (up to 255 characters) to be
   processed by NetView for this operator when this operator logs on to
   NetView.

   If the command or command list you specify contains any commas, blanks,
   or other special characters that TSO/E requires to be quoted, it must be
   enclosed in single quotation marks.

MSGRECVR(NO | YES)
   Specifies whether this operator is to receive unsolicited messages that are
   not routed to a specific NetView operator.

   NO      Specifies that the operator is not to receive the messages.
           NO is the default.

   YES     Specifies that the operator is to receive the messages.

NGMFADMN(NO | YES)
   Specifies whether a NetView operator has administrator authority to the
   NetView Graphic Monitor Facility (NGMF).

   NO      Specifies that the operator does not have authority.
           NO is the default.

   YES     Specifies that the operator has the authority.

NGMFVSPN(view-span)
   Reserved for future use by the NetView Graphic Monitor Facility.

OPCLASS(class ...)
   NetView scope classes for which the operator has authority. The OPCLASS
   values are only used if NetView is doing the checking itself, rather than
   using SAF and the NETCMDS class that RACF provides. If the OPCLASS
   operand is not specified, the operator is considered to have authority in
   scope classes.

   The class value is a number from 1 to 2040 that specifies a NetView scope
class.
ADDUSER

OIDCARD | NOOIDCARD

OIDCARD
Specifies that the new user must supply an operator identification card when logging onto the system. If you specify the OIDCARD operand, the system prompts you to enter the new user's operator identification card as part of the processing of the ADDUSER command. If you specify the OIDCARD operand in a job executing in the background or when you cannot be prompted in the foreground, the ADDUSER command fails.

NOOIDCARD
Specifies that the new user is not required to supply an operator identification card. NO_OIDCARD is the default value if you omit both OIDCARD and NOOIDCARD.

OMVS
Specifies z/OS UNIX System Services information for the user being defined to RACF. Information is stored in the OMVS segment of the user's profile.

You can control access to an entire OMVS segment or to individual fields in the OMVS segment by using field-level access checking.

ASSIZEMAX(address-space-size)
Specifies the RLIMIT_AS hard limit (maximum) resource value that processes receive when they are dubbed a process. The address-space-size you define to RACF is a numeric value from 10485760–2 147 483 647. ASSIZEMAX indicates the address space region size in bytes. The soft limit (current) resource value is obtained from MVS. If the soft limit value from MVS is greater than the address-space-size, the soft limit is used.

The value specified for ASSIZEMAX is also used when processes are initiated by a daemon process using an exec after setuid(). In this case, both the RLIMIT_AS hard and soft limits are set to the address-space-size value.

The value specified for ASSIZEMAX overrides any value provided by the MAXASSIZE parameter of BPXPRMxx. For more information, see z/OS UNIX System Services Planning.

AUTOUID | UID
Specifies whether RACF is to automatically assign an unused UID value to the user or if a specific UID value is to be assigned.

AUTOUID
Specifies that RACF is to automatically assign an unused UID value to the user. The UID value is derived from information obtained from the BPX.NEXT.USER profile in the FACILITY class. For more information on setting up BPX.NEXT.USER, see z/OS Security Server RACF Security Administrator’s Guide.

If you are using RRSF automatic command direction for the USER class, the command sent to other nodes will contain an explicit assignment of the UID value which was derived by RACF on the local node.

Rules:
- AUTOUID cannot be specified if more than one user ID is entered.
- The AUTOUID keyword is mutually exclusive with the SHARED keyword.
- If both UID and AUTOUID are specified, AUTOUID is ignored.
Field-level access checking for the UID field applies when using AUTOUID.

**UID**(user-identifier) [SHARED]

UID(user-identifier)

Specifies the user identifier. The UID is a numeric value from 0–2 147 483 647.

When assigning a UID to a user, you should make sure that the user's default group has a GID. A user who has a UID and a current connect group that has a GID can use functions such as the TSO/E OMVS command and can access z/OS UNIX files based on the UID and GID values assigned.

Care should be taken in assigning 0 as the user identifier. UID 0 is considered a superuser. The superuser passes all z/OS UNIX security checks. Assigning a UID to a user ID that appears in the RACF started procedures table (ICHRIN03) should also be done with care. RACF defined started tasks that have the trusted or privileged attribute are considered superusers even if their UID is a value other than 0.

**Rules:**

- If the security administrator has defined the SHARED.IDS profile in the UNIXPRIV class, the UID value must be unique. Use the SHARED keyword in addition to UID to assign a value that is already in use.
- If SHARED.IDS is not defined, RACF does not require the UID to be unique. The same value can be assigned to multiple users but this is not recommended because individual user control would be lost. However, if you want a set of users to have exactly the same access to z/OS UNIX resources, you might decide to assign the same UID to more than one user.
- The maximum number of user IDs that can share a UID or groups that can share a GID is 132 when each consists of 8 characters. More user IDs or groups are available using less than 8 characters. If the limit is met, you can combine user ID functions (for started tasks or daemons) to use physically less user IDs sharing the same UID. You may also use SUPERUSER granularity functionality to reduce the need to assign and share SUPERUSER authority using UID 0.
- If the UID is not specified, the user is unable to become a z/OS UNIX user and a LISTUSER for that user ID shows NONE for the UID.

**SHARE**

If the security administrator has chosen to control the use of shared UIDs, this keyword must be used in addition to the UID keyword to specify the user identifier if it is already in use by at least one other user. The administrator controls shared UIDs by defining the SHARED.IDS profile in the UNIXPRIV class.

**Rules:**

- If the SHARED.IDS profile is not defined, SHARED is ignored.
- If SHARED is specified in the absence of UID, it is ignored.
ADDUSER

- If the SHARED.IDS profile is defined and SHARED is specified, but the value specified with UID is not currently in use, SHARED is ignored and UNIXPRIV authority is not required.
- Field-level access checking for the UID field applies when using SHARED.
- The SHARED keyword is mutually exclusive with the AUTOUID keyword.

CPUTIMEMAX(*cpu-time*)
Specifies the RLIMIT_CPU hard limit (maximum) resource value that the user’s z/OS UNIX processes receive when they are dubbed a process. The *cpu-time* you define to RACF is a numeric value from 7–2,147,483,647. RLIMIT_CPU indicates the *cpu-time* in seconds that a process is allowed to use. The soft limit (current) resource value is obtained from MVS. If the soft limit value from MVS is greater than the *cpu-time* value, the soft limit is used.

The value specified for CPUTIMEMAX is also used when processes are initiated by a daemon process using an exec after setuid(). In this case, both the RLIMIT_CPU hard limit and the soft limit are set to the *cpu-time* value.

For processes running in, or forked from TSO or BATCH, the *cpu-time* value has no effect. For processes created by the rlogin command or other daemons, the *cpu-time* is the time limit for the address space.

The value specified for CPUTIMEMAX overrides any value provided by the MAXCPUTIME parameter of BPXPRMxx. For more information, see [z/OS UNIX System Services Planning](https://www.ibm.com/support/knowledgecenter/en/SSGEG2_1.11.0/com.ibm.zos.v1r11.racf.ada.1112/adaman/adamanu.html).

FILEPROCMAX(*files-per-process*)
Specifies the maximum number of files this user is allowed to have concurrently active or open. The *files-per-process* you define to RACF is a numeric value from 3 and 524,287. FILEPROCMAX is the same as the OPEN_MAX variable defined in the POSIX standard.

FILEPROCMAX lets you limit the amount of system resources available to a user process. Select FILEPROCMAX by considering:
- For conformance to standards, set FILEPROCMAX to:
  - At least 16 to conform to the POSIX standard, and
  - At least 25 to conform to the FIPS standard.
- 256 is a commonly recommended value.
- A process can change its own value for the number of files it has active or open using the setrlimit() function. Only processes with appropriate privileges can increase their limits.
- The minimum value of 3 supports the standard files for a process: standard input, standard output, and standard error.
- The value needs to be larger than 3 to support z/OS UNIX shell users. If the value is too small, the shell might issue the message, File descriptor not available.

The value specified for FILEPROCMAX overrides any value provided by the MAXFILEPROC parameter of BPXPRMxx. For more information, see [z/OS UNIX System Services Planning](https://www.ibm.com/support/knowledgecenter/en/SSGEG2_1.11.0/com.ibm.zos.v1r11.racf.ada.1112/adaman/adamanu.html).
HOME(*initial-directory-name*)
Specifies the user’s z/OS UNIX initial directory pathname. This is the current working directory for the user’s process when the user enters the TSO/E.

When you define a HOME directory name to RACF, it can contain 1–1023 characters. The HOME pathname can consist of any characters and can be entered with or without single quotation marks. The following rules apply:

- If parentheses, commas, blanks, or semicolons are to be entered as part of the pathname, the character string must be enclosed in single quotation marks.
- If a single quotation mark is intended to be part of the pathname, use two single quotation marks together for each single quotation mark within the string, and enclose the entire string within single quotation marks.

Both uppercase and lowercase characters are accepted and maintained in the case in which they are entered. The fully qualified pathname should be specified. RACF does not ensure that a valid pathname has been specified.

If you issue the ADDUSER command as a RACF operator command and you specify the pathname in lowercase, you must include the pathname within single quotations.

If HOME is not specified, MVS sets the working directory for the user to / (the root directory). However, the default value is not placed in the user’s profile, and is not displayed when a LISTUSER command is entered.

MEMLIMIT | NOMEMLIMIT

MEMLIMIT(*nonshared-memory-size*)
Specifies the maximum number of bytes of nonshared memory that can be allocated by the user. The nonshared-memory-size value must be numeric 0–16777215, followed by the letter M, G, T, or P. The M, G, T or P letter indicates the multiplier to be used. The maximum value is 16383P.

<table>
<thead>
<tr>
<th></th>
<th>Decimal</th>
<th>Binary</th>
<th>Hexadecimal</th>
</tr>
</thead>
<tbody>
<tr>
<td>M—megabyte</td>
<td>1048576</td>
<td>2^20</td>
<td>00000000 00100000</td>
</tr>
<tr>
<td>G—gigabyte</td>
<td>1073741824</td>
<td>2^30</td>
<td>00000000 40000000</td>
</tr>
<tr>
<td>T—terabyte</td>
<td>1099511627776</td>
<td>2^40</td>
<td>00000000 00000000</td>
</tr>
<tr>
<td>P—petabyte</td>
<td>1125899906842624</td>
<td>2^50</td>
<td>00000000 00000000</td>
</tr>
</tbody>
</table>

The following are different MEMLIMIT(*nonshared-memory-size*) examples:
- MEMLIMIT(1M) indicates a nonshared-memory-size of 1048576 bytes.
- MEMLIMIT(1500M) indicates a nonshared-memory-size of 1572864000 bytes.
- MEMLIMIT(10G) indicates a nonshared-memory-size of 10737418240 bytes.

For more extensive information, see z/OS UNIX System Services Planning.

NOMEMLIMIT
Specifies that you want to delete the nonshared memory size from the OMVS segment of the user’s profile.

MMAPAREAMAX(*memory-map-size*)
Specifies the maximum amount of data space storage, in pages, that can
be allocated by the user for memory mappings of z/OS UNIX files. Storage is not allocated until memory mappings are active. The *memory-map-size* you define to RACF is a numeric value from 1–16777216.

Use of memory map services consumes a significant amount of system memory. For each page (4KB) that is memory mapped, 96 bytes of ESQA are consumed when a file is not shared with any other users. When a file is shared by multiple users, each subsequent user after the initial user causes 32 bytes of ESQA to be consumed for each shared page. The ESQA storage is consumed when the *mmap()* function is invoked by the application program.

The value specified for MMAPAREAMAX overrides any value provided by the MAXMMAPAREA parameter of BPXPRMxx. For more information, see z/OS UNIX System Services Planning.

**PROCUSERMAX** (*processes-per-UID*)

Specifies the maximum number of processes this user is allowed to have active at the same time, regardless of how the process became a z/OS UNIX process. The *processes-per-UID* you define to RACF is a numeric value from 3–32767. PROCUSERMAX is the same as the CHILD_MAX variable defined in the POSIX standard.

PROCUSERMAX allows you to limit user activity to optimize performance. Select PROCUSERMAX by considering:

- For conformance to standards, set PROCUSERMAX to:
  - At least 16 to conform to the POSIX standard, and
  - At least 25 to conform to the FIPS standard.
- A user with a UID of 0 is not limited by the PROCUSERMAX value because a superuser might need to be capable of logging on and using z/OS UNIX services to solve a problem.
- A low PROCUSERMAX value limits the number of concurrent processes that the user can run. A low value also limits the user's consumption of processing time, virtual storage, and other system resources.
- Some daemons run without UID 0, and might create many address spaces. In these cases, it is necessary to set the limit high enough for the daemon associated with this user ID to run all of its processes.

Though not recommended, the same OMVS UID can be given to more than one user ID. If users share a UID, you need to define a greater number for PROCUSERMAX. An example is the user ID defined for the default OMVS segment, specified by the FACILITY class profile BPX.DEFAULT.USER.

The value specified for PROCUSERMAX overrides any value provided by the MAXPROCUSER parameter of BPXPRMxx. For more information, see z/OS UNIX System Services Planning.

**PROGRAM**( *program-name*)

Specifies the PROGRAM pathname (z/OS UNIX shell program). This is the first program started when the TSO/E command OMVS is entered or when a batch job is started using the BPXBATCH program.

When you define a PROGRAM pathname to RACF, it can contain 1–1023 characters. The PROGRAM pathname can consist of any characters and can be entered with or without single quotation marks. The following rules apply:
ADDUSER

- If parentheses, commas, blanks, or semicolons are to be entered as part of the pathname, the character string must be enclosed in single quotation marks.
- If a single quotation mark is intended to be part of the pathname, use two single quotation marks together for each single quotation mark within the string, and enclose the entire string within single quotation marks.

Both uppercase and lowercase characters are accepted and maintained in the case in which they are entered. The fully qualified pathname should be specified. RACF does not ensure that a valid pathname has been specified. If you issue the ADDUSER command as a RACF operator command and you specify the pathname in lowercase, you must include the pathname within single quotations.

If PROGRAM is not specified or if PROGRAM is specified as blanks, MVS gives control to the default z/OS UNIX shell program. However, the default value is not placed in the user's profile, and is not displayed when a LISTUSER command is entered.

For more information about the default z/OS UNIX shell program supplied with z/OS UNIX, see z/OS UNIX System Services Planning and z/OS UNIX System Services User's Guide.

SHMEMMAX | NOSHMEMMAX

**SHMEMMAX(shared-memory-size)**

Specifies the maximum number of bytes of shared memory that can be allocated by the user. The shared-memory-size value must be numeric 1–16777215, followed by the letter M, G, T, or P. The M, G, T or P letter indicates the multiplier to be used. The maximum value is 16383P.

<table>
<thead>
<tr>
<th></th>
<th>Decimal</th>
<th>Binary</th>
<th>Hexadecimal</th>
</tr>
</thead>
<tbody>
<tr>
<td>M—megabyte</td>
<td>1048576</td>
<td>2^20</td>
<td>00000000 00100000</td>
</tr>
<tr>
<td>G—gigabyte</td>
<td>1073741824</td>
<td>2^30</td>
<td>00000000 40000000</td>
</tr>
<tr>
<td>T—terabyte</td>
<td>109951162776</td>
<td>2^40</td>
<td>00000100 00000000</td>
</tr>
<tr>
<td>P—petabyte</td>
<td>1125899906842624</td>
<td>2^50</td>
<td>00040000 00000000</td>
</tr>
</tbody>
</table>

The following are different SHMEMMAX(shared-memory-size) examples:
- **SHMEMMAX(1M)** indicates a shared-memory-size of 1048576 bytes.
- **SHMEMMAX(1500M)** indicates a shared-memory-size of 1572864000 bytes.
- **SHMEMMAX(1G)** indicates a shared-memory-size of 10737418240 bytes.

The value specified for SHMEMMAX overrides any value provided by the IPCSHMMPPAGES parameter of BPXPRMxx. For more information, see z/OS UNIX System Services Planning.

**NOSHMEMMAX**

Specifies that you want to delete the shared memory size from the OMVS segment of the user's profile. The value specified for IPCSHMMPPAGES in BPXPRMxx now applies to the user.

**THREADSMAX(threads-per-process)**

Specifies the maximum number of pthread_create threads, including those running, queued, and exited but not detached, that this user can have concurrently active. The threads-per-process you define to RACF is a
numeric value from 0–100000. Specifying a value of 0 prevents applications run by this user from using the pthread_create service.

The value specified for THREADSMAX overrides any value provided by the MAXTHREADS parameter of BPXPRMxx. For more information, see z/OS UNIX System Services Planning.

**OPERATIONS | NOOPERATIONS**

**OPERATIONS**
Specifies that the new user has authorization to do maintenance operations on all RACF-protected data sets, tape volumes, and DASD volumes except those where the access list specifically limits the OPERATIONS user to a lower access authority than the operation requires.

The OPERATIONS attribute allows the user to access VM resources except those where the resource's access list specifically limits the OPERATIONS user to a lower access authority.

You establish the lower access authority for the OPERATIONS user through the PERMIT command. OPERATIONS specified on ADDUSER overrides NOOPERATIONS specified on the CONNECT command.

You must have the SPECIAL attribute to enter the OPERATIONS operand.

**NOOPERATIONS**
Specifies that the new user is not to have the OPERATIONS attribute.
NOOPERATIONS is the default if you omit both OPERATIONS and NOOPERATIONS.

**OPERPARM**
Specifies default information used when this user establishes an extended MCS console session.

You can control access to the entire OPERPARM segment or to individual fields within the OPERPARM segment by using field-level access checking. For more information, see z/OS Security Server RACF Security Administrator’s Guide.

For information on planning how to use OPERPARM segments, see z/OS MVS Planning: Operations.

**Notes:**
1. You need not specify every suboperand in an OPERPARM segment. In general, if you omit a suboperand, the default is the same as the default in the CONSOLxx PARMLIB member, which can also be used to define consoles.
2. If you specify MSCOPE or ROUTCODE but do not specify a value for them, RACF uses MSCOPE(*ALL) and ROUTCODE(NONE) to update the corresponding fields in the user profile, and these values appear in listings of the OPERPARM segment of the user profile.
3. If you omit the other suboperands, RACF does not update the corresponding fields in the user's profile, and no value appears in listings of the OPERPARM segment of the profile.

**ALTGRP(alternate-console-group)**
Specifies the console group used in recovery. It can contain 1–8 characters. Valid characters are 0–9, A–Z, # (X'7B'), $ (X'5B'), or @ (X'7C').

Restriction: Starting with z/OS Version 1 Release 8, console services ignores ALTGRP(alternate-console-group) when a session is established and it need not be specified.
**ADDUSER**

**AUTH** Specifies the authority this console has to issue operator commands.

If you omit this operand, RACF does not add this field to the user’s profile. However, an extended MCS console uses AUTH(INFO) when a session is established.

**MASTER**

Allows this console to act as a master console, which can issue all MVS operator commands.

**ALL**

Allows this console to issue system control commands, input/output commands, console control commands, and informational commands.

**INFO**

Allows this console to issue informational commands.

**CONS**

Allows this console to issue console control and informational commands.

**IO**

Allows this console to issue input/output and informational commands.

**SYS**

Allows this console to issue system control commands and informational commands.

**AUTO(YES | NO)**

Specifies whether the extended console can receive messages that have been automated by the Message Processing Facility (MPF) in the sysplex.

If you omit this operand, RACF does not add this field to the user’s profile. However, an extended MCS console uses AUTO(NO) when a session is established.

**CMDSYS**(system-name | *)

Specifies the system to which commands issued from this console are to be sent. The system-name value must be 1–8 characters. Valid characters are A–Z, 0–9, @ (X’7C’), # (X’7B’), and $ (X’5B’). If * is specified, commands are processed on the local system where the console is attached.

If you omit this operand, RACF does not add this field to the user’s profile. However, an extended MCS console uses CMDSYS(*) when a session is established.

**DOM**

Specifies whether this console receives delete operator message (DOM) requests.

If you omit this operand, RACF does not add this field to the user’s profile. However, an extended MCS console uses DOM(NORMAL) when a session is established.

**NORMAL**

 Specifies that the system queues all appropriate DOM requests to this console.

**ALL**

Specifies that all systems in the sysplex queue DOM requests to this console.
NONE
Specifies that no DOM requests are queued to this console.

HC(YES | NO)
Specifies whether this console is to receive hardcopy messages.
If you omit this operand, RACF does not add this field to the user’s profile. However, z/OS console services uses HC(NO) when a session is established.

INTIDS(YES | NO)
Specifies whether this console is to receive messages directed to console ID zero (the internal console). Such messages are usually responses to internally issued commands.
If you omit this operand, RACF does not add this field to the user’s profile. However, z/OS console services will use INTIDS(NO) when a session is established.

KEY(searching-key)
Specifies a 1–8 byte character name that can be used to display information for all consoles with the specified key by using the MVS command DISPLAY CONSOLES,KEY. If specified, KEY can include A–Z, 0–9, # (X’7B’), $ (X’5B’), or @ (X’7C’).
If you omit this operand, RACF does not add this field to the user’s profile. However, an extended MCS console uses a KEY value of NONE when a session is established.

LEVEL
Specifies the messages that this console is to receive. The message-level variable can be a list of R, I, CE, E, IN, NB or ALL. If you specify ALL, you cannot specify R, I, CE, E, or IN.
If you omit this operand, RACF does not add this field to the user’s profile. However, an extended MCS console uses LEVEL(ALL) when a session is established.

NB The console receives no broadcast messages.
ALL The console receives these messages: R, I, CE, E, IN.
R The console receives messages requiring an operator reply.
I The console receives immediate action messages.
CE The console receives critical eventual action messages.
E The console receives eventual action messages.
IN The console receives informational messages.

LOGCMDRESP
Specifies if command responses are to be logged.
If you omit this operand, RACF does not add this field to the user’s profile. However, an extended MCS console uses LOGCMDRESP(SYSTEM) when a session is established.

SYSTEM
Specifies that command responses are logged in the hardcopy log.

NO
Specifies that command responses are not logged.
MFORM(message-format)
Specifies the format in which messages are displayed at the console.
Can be a combination of J, M, S, T, and X:

J Messages are displayed with a job ID or name.
M Message text is displayed.
S Messages are displayed with the name of the originating system.
T Messages are displayed with a time stamp.
X Messages that are flagged as exempt from job name and system name formatting are ignored.

If you omit this operand, RACF does not add this field to the user’s profile. However, an extended MCS console uses MFORM(M) when a session is established.

MIGID(YES I NO)
Specifies that a 1-byte migration ID is to be assigned to this console.
The migration ID allows command processors that use a 1-byte console ID to direct command responses to this console.

Restriction: Starting with z/OS Version 1 Release 7, console services ignores MIGID(YES I NO) when a session is established and it need not be specified.

MONITOR(events)
Specifies which information should be displayed when jobs, TSO sessions, or data set status are being monitored.

If you omit this operand, RACF does not add this field to the user’s profile. However, an extended MCS console uses MONITOR(JOBNAMES SESS) when a session is established. The events value can be a list of the following:

JOBNAMES | JOBNAMEST
Displays information about the start and end of each job.
JOBNAMES omits the times of job start and job end. JOBNAMEST displays the times of job start and job end.

SESS | SESST
Displays information about the start and end of each TSO session.
SESS omits the times of session start and session end. SESST displays them.

STATUS
Specifies that the information displayed when a data set is freed or unallocated should include the data set status.

MSCOPE
Specifies the systems from which this console can receive messages that are not directed to a specific console.

If you omit this operand, RACF does not add this field to the user’s profile. However, an extended MCS console uses MSCOPE(*ALL) when a session is established.

If you specify MSCOPE but omit a value, RACF uses MSCOPE(*ALL) to update this field in the user’s profile. *ALL appears in listings of the OPERPARM segment of the user’s profile.
ADDUSER

system-name
Is a list of one or more system names, where system-name can be any combination of A–Z, 0–9, # (X'7B'), $ (X'5B'), or @ (X'7C').

* Is the system on which the console is currently active.

*ALL
All systems.

ROUTCODE(ALL | NONE | routing-codes)
Specifies the routing codes of messages this console is to receive.

If you omit this operand, RACF does not add this field to the user's profile. However, an extended MCS console uses ROUTCODE(NONE) when a session is established.

If you specify ROUTCODE but omit a value, RACF uses ROUTCODE(NONE) to update this field in the user's profile. NONE appears in listings of the OPERPARM segment of the user's profile. The value for ROUTCODE can be one of the following:

ALL
All routing codes.

NONE
No routing codes.

routing-codes
One or more routing codes or sequences of routing codes. The routing codes can be list of \( n \) and \( n1:n2 \), where \( n, n1, \) and \( n2 \) are integers 1–128, and \( n2 \) is greater than \( n1 \).

STORAGE(amount)
Specifies the amount of storage in megabytes in the TSO/E user's address space that can be used for message queuing to this console. If specified, STORAGE must be a number from 1–2000.

If you omit this operand, RACF does not add this field to the user's profile. However, an extended MCS console uses STORAGE(1) when a session is established and a value of 0 is listed in the OPERPARM segment of the user's profile to indicate that no storage value was specified.

UD(YES | NO)
Specifies whether this console is to receive undelivered messages. If you omit this operand, RACF does not add this field to the user's profile.

Restriction: Starting with z/OS Version 1 Release 8, console services ignores UD(YES | NO) when a session is established and it need not be specified.

UNKNIDS(YES | NO)
Specifies whether this console is to receive messages directed to unknown console IDs. Unknown consoles are typically one-byte console IDs that the system cannot unambiguously resolve.

If you omit this operand, RACF does not add this field to the user's profile. However, z/OS console services will use UNKNIDS(NO) when a session is established.
OVM
Specifies OpenExtensions VM information for the user being defined.
Information is stored in the OVM segment of the user’s profile.
You can control access to an entire OVM segment or to individual fields within
the OVM segment by using field level access checking.

FSROOT(file-system-root)
Specifies the pathname for the file system root.
When you define the FSROOT pathname to RACF, it can contain 1–1023
characters, consist of any character, and be entered with or without single
quotation marks. The following rules apply:
• If parentheses, commas, blanks, or semicolons are entered as part of the
  pathname, the character string must be enclosed in single quotation
  marks. For example if the pathname is (123), you must enter
  FSROOT('123').
• If a single quotation mark is intended to be part of the pathname, use
two single quotation marks together for each single quotation mark within
the string, and enclose the entire string within single quotation marks.
When entering the ADDUSER command, both uppercase and lowercase
characters are accepted and maintained in the case in which they are
entered.
If you do not specify a value for FSROOT in the OVM segment, VM uses
the value specified in the CP directory. If no value is specified in the CP
directory, issue the OPENVM MOUNT command to mount the appropriate
file system.

HOME(initial-directory-name)
Specifies the initial directory pathname. The initial directory is part of the file
system and is the current working directory for the user’s process when the
user enters the OPENVM SHELL can contain 1–1023 characters, consist of
any character, and be entered with or without single quotation marks. The
following rules apply:
• If parentheses, commas, blanks, or semicolons are entered as part of the
  pathname, the character string must be enclosed in single quotation
  marks. For example if the pathname is (123), you must enter
  HOME('123').
• If a single quotation mark is intended to be part of the pathname, use
two single quotation marks together for each single quotation mark within
the string, and enclose the entire string within single quotation marks.
When entering the ADDUSER command, both uppercase and lowercase
characters are accepted and maintained in the case in which they are
entered.
If no value is specified for HOME in the OVM segment, VM uses the value
specified in the CP directory. If no value is specified in the CP directory, VM
sets the working directory for the user to /, the root directory.

PROGRAM(program-name)
Specifies the PROGRAM pathname (z/OS UNIX shell program). This is the
first program started when the OPENVM SHELL command is entered.
When you define a PROGRAM pathname to RACF, it can contain 1–1023
characters, consist of any character and be entered with or without single
quotation marks. The following rules apply:
ADDUSER

- If parentheses, commas, blanks, or semicolons are entered as part of the pathname, the character string must be enclosed in single quotation marks. For example if the pathname is (123), you must enter PROGRAM('(123)').
- If a single quotation mark is intended to be part of the pathname, use two single quotation marks together for each single quotation mark within the string, and enclose the entire string within single quotation marks.

When entering the ADDUSER command for OVM segment information, both uppercase and lowercase characters are accepted and maintained in the case in which they are entered. Specify the fully qualified pathname, because RACF does not ensure that a valid pathname has been specified.

If no value is specified for PROGRAM in the OVM segment, VM uses the value specified in the CP directory. If no value is specified in the CP directory, VM gives control to the default z/OS UNIX shell program (/bin/sh) when a user issues the OPENVM SHELL command.

UID(user-identifier)
Specifies the user identifier. The UID is a numeric value from 0–2 147 483 647.

Care should be taken in assigning 0 as the user identifier. UID 0 is considered a superuser.

If UID is not specified, the user is assigned the default UID of 4294967295 (X'FFFFFFFF') and the LISTUSER command for that user ID shows NONE for the UID.

Note: RACF does not require the UID to be unique. You can assign the same value to multiple users, but this is not recommended because individual user control is lost. However, if you want a set of users to have exactly the same access to the OpenExtensions VM resources, you can assign the same UID to more than one user.

OWNER(userid or group-name)
Specifies a RACF-defined user or group to be assigned as the owner of the RACF profile for the user being added. If you omit this operand, you are defined as the owner.

PASSWORD | NOPASSWORD

PASSWORD(password)
Specifies the user's initial logon password. This password is always set expired, thus requiring the user to change the password at initial logon. Note that the password syntax rules your installation defines using SETROPTS PASSWORD do not apply to this password.

If you omit both PASSWORD and NOPASSWORD, or enter PASSWORD with no value, RACF takes the group name from the DFLTGRP operand as the default password.

NOPASSWORD
Specifies that the new user does not need to supply an initial logon password when first entering the system if OIDCARD is also specified. If you specify NOOIDDACARD (or you allow this option to default) and you specify NOPASSWORD, you define a protected user ID that cannot be used to enter the system by any means that requires a password to be specified, such as a TSO logon, CICS signon, or batch job that specifies a password on the JOB statement. Therefore, user IDs that you assign to z/OS UNIX, UNIX daemons, started procedures, applications, servers or...
subsystems can be protected from being revoked when an incorrect password is entered. If the user attempts to enter the system with a password, the attempt fails. Note that the protected user ID is not revoked due to the failed password attempts even if the SETROPTS PASSWORD(REVOKE) option is in effect.

Determine which user IDs you wish to protect, ensuring that these user IDs will not be used in any circumstance where a password must be supplied. A protected user will have the PROTECTED attribute displayed in the output of the LISTUSER command. Protected users can be associated with started procedures defined in the STARTED class (preferred method) or in the started procedures table (ICHRIN03).

**Note:** Kerberos information, such as a local principal name, must not be defined for protected user IDs and these user IDs must not be used for Kerberos authentication, since Kerberos authentication failures can result in user revocation.

**PHRASE('password-phrase')**

Specifies the user’s initial password phrase. The password phrase you define is a text string of up to 100 characters and must be enclosed in single quotation marks. The password phrase is always set expired, thus requiring the user to change it on initial use.

When the new-password-phrase exit (ICHPWX11) is present and allows it, the password phrase can be 9–100 characters. When ICHPWX11 is not present, the password phrase must be 14–100 characters. Contact your system programmer to find out if your installation uses the new-password-phrase exit (ICHPWX11) or see [z/OS Security Server RACF System Programmer’s Guide](#) for programming details.

Every user that you assign a password phrase must have a password. When you specify PHRASE for a user without specifying PASSWORD, the user is assigned the default password. When you specify PHRASE with NOPASSWORD, an error message is issued indicating that the NOPASSWORD operand is ignored and the user is assigned the default password.

The following syntax rules apply to all password phrases. You cannot alter these syntax rules but you can specify additional syntax rules if your installation tailors the new-password-phrase exit (ICHPWX11).

**Syntax rules for password phrases:**

- Maximum length: 100 characters
- Minimum length:
  - 9 characters, when ICHPWX11 is present and allows the new value
  - 14 characters, when ICHPWX11 is not present
- Must not contain the user ID (as sequential uppercase or sequential lowercase characters)
- Must contain at least 2 alphabetic characters (A–Z, a–z)
- Must contain at least 2 non-alphabetic characters (numerics, punctuation, or special characters)
- Must not contain more than 2 consecutive characters that are identical
- If a single quotation mark is intended to be part of the password phrase, you must use two single quotation marks together for each single quotation mark.

If the new-password-phrase exit (ICHPWX11) is present, it can reject the specified password phrase. RACF rejects password phrases shorter than 14 characters unless ICHPWX11 is present and allows the new value.
ADDUSER

If the specified password phrase is accepted, it is made the user's current password phrase and, when SETROPTS PASSWORD(HISTORY) is in effect, it is added to the user's password phrase history.

If you omit PHRASE, no password phrase is assigned. If you enter PHRASE without a password-phrase value, you are prompted for a value unless your TSO session is in NOPROMPT mode.

PROXY

Specifies information which the z/OS LDAP server will use when acting as a proxy on behalf of a requester. The **R_proxyserv** (IRRSPY00) SAF callable service will attempt to retrieve this information when it is not explicitly supplied with the invocation parameters. Applications or other services which use the **R_proxyserv** callable service, such as IBM Policy Director Authorization Services for z/OS and OS/390, may instruct their invokers to define PROXY segment information.

**LDAPHOST(ldap_url)**

Specifies the URL of the LDAP server which the z/OS LDAP server will contact when acting as a proxy on behalf of a requester. An LDAP URL has a format such as `ldap://123.45.6:389` or `ldaps://123.45.6:636`, where `ldaps` indicates that an SSL connection is desired for a higher level of security. LDAP will also allow you to specify the host name portion of the URL using either the text form (BIGHOST.POK.IBM.COM) or the dotted decimal address (123.45.6). The port number is appended to the host name, separated by a colon (`X'7A'`).


The LDAP URL that you define to RACF can consist of 10–1023 characters. A valid URL must start with either `ldap://` or `ldaps://`. RACF will allow any characters to be entered for the remaining portion of the URL, but you should ensure that the URL conforms to TCP/IP conventions. For example, parentheses, commas, blanks, semicolons, and single quotation marks are not typically allowed in a host name. The LDAP URL can be entered with or without single quotation marks, however, in both cases, it will be translated to uppercase.

RACF does not ensure that a valid LDAP URL has been specified.

**BINDDN(bind_distinguished_name)**

Specifies the distinguished name (DN) which the z/OS LDAP server will use when acting as a proxy on behalf of a requester. This DN will be used in conjunction with the BIND password if the z/OS LDAP server needs to supply an administrator or user identity to BIND with another LDAP server. A DN is made up of attribute value pairs, separated by commas. For example:

- `cn=Ben Gray,ou=editing,o=New York Times,c=US`
- `cn=Lucille White,ou=editing,o=New York Times,c=US`
- `cn=Tom Brown,ou=reporting,o=New York Times,c=US`

When you define a BIND DN to RACF, it can contain 1–1023 characters. The BIND DN can consist of any characters and can be entered with or without single quotation marks. The following rules apply:
ADDUSER

- If parentheses, commas, blanks, or semicolons are to be entered as part of the BIND DN, the character string must be enclosed in single quotation marks.
- If a single quotation mark is intended to be part of the BIND DN, use two single quotation marks together for each single quotation mark within the string, and enclose the entire string within single quotation marks.

Both uppercase and lowercase characters are accepted and maintained in the case in which they are entered. For more information about LDAP distinguished names, see IBM Tivoli Directory Server Administration and Use for z/OS or z/OS Integrated Security Services LDAP Server Administration and Use.

If you issue the ADDUSER command as a RACF operator command and you specify the BIND DN in lowercase, you must include the BIND DN within single quotations.

RACF does not ensure that a valid BIND DN has been specified.

BINDPW

Specifies the password which the z/OS LDAP server will use when acting as a proxy on behalf of a requester.

When you define a BIND password to RACF, it can contain 1–128 characters. The BIND password can consist of any characters (see exception below) and can be entered with or without single quotation marks.

Rules:
- The BIND password cannot start with the left brace { character (X'8B').
- If parentheses, commas, blanks, or semicolons are to be entered as part of the BIND password, the character string must be enclosed in single quotation marks.
- If a single quotation mark is intended to be part of the BIND password, use two single quotation marks together for each single quotation mark within the string, and enclose the entire string within single quotation marks.

Both uppercase and lowercase characters are accepted and maintained in the case in which they are entered. For more information about LDAP passwords, see IBM Tivoli Directory Server Administration and Use for z/OS or z/OS Integrated Security Services LDAP Server Administration and Use.

If you issue the ADDUSER command as a RACF operator command and you specify the BIND password in lowercase, you must include the BIND password within single quotations.

RACF does not ensure that a valid BIND password has been specified.
Attention:

- When the command is issued from ISPF, the TSO command buffer (including possible BINDPW password data) is written to the ISPLOG data set. As a result, you should not issue this command from ISPF or you must control the ISPLOG data set carefully.
- When the command is issued as a RACF operator command, the command and the possible BINDPW password data is written to the system log. Therefore, use of ADDUSER as a RACF operator command should either be controlled or you should issue the command as a TSO command.

**RESTRICTED | NORESTRICTED**

**RESTRICTED**

Specifies that global access checking is bypassed when resource access checking is performed for the new user, and neither ID(*) on the access list nor the UACC will allow access. The RESTRICTED.FILESYS.ACCESS profile in the UNIXPRIV class can also be used to bypass the z/OS UNIX 'other' permission bits during file access checking for RESTRICTED users.

**Note:** If your installation has profiles defined in the PROGRAM class and the user ID with the RESTRICTED attribute needs to load programs covered by one or more of these profiles, the user ID or a group to which the user is connected must be put on the access list with EXECUTE or READ authority.

**NORESTRICTED**

Specifies that the new user does not have the RESTRICTED attribute and access checking is performed the standard way including global access checking, ID(*), the UACC, and the z/OS UNIX 'other' permission bits as appropriate. NORESTRICTED is the default value if you omit both the RESTRICTED and NORESTRICTED keywords.

**SECLABEL(security-label)**

Specifies the user's default security label, where security-label is an installation-defined security label name that represents an association between a particular security level and zero or more security categories.

If the user does not enter a security label when entering the system, and none is assigned based on the user's port of entry, this value becomes the user's current security label.

A security label corresponds to a particular security level (such as CONFIDENTIAL) with a set of zero or more security categories (such as PAYROLL or PERSONNEL).

When no profile exists in the SECLABEL class for security-label, the ADDUSER command fails and the user is not added.

**SECLEVEL(security-level)**

Specifies the user's security level, where security-level is an installation-defined security level name that must be a member of the SECLEVEL profile in the SECDATA class. The security-level that you specify corresponds to the number of the minimum security level that a user must have to access the resource.

When you specify SECLEVEL and the SECDATA class is active, RACF adds security level access checking to its other authorization checking. If global access checking does not grant access, RACF compares the security level allowed in the user profile with the security level required in the resource profile. If the security level in the user profile is less than the security level in the
resource profile, RACF denies the access. If the security level in the user profile is equal to or greater than the security level in the resource profile, RACF continues with other authorization checking.

**Note:** RACF does not perform security level checking for a started task or user that has the RACF privileged or trusted attribute. The RACF privileged or trusted attribute can be assigned to a started task through the RACF started procedures table or STARTED class, or to other users by installation-supplied RACF exits.

When the SECDATA class is not active, RACF ignores this operand. When no member of the SECELEVEL profile exists for security-level, you are prompted to provide a valid security-level.

### SPECIAL | NOSPECIAL

**SPECIAL**

Specifies that the new user is allowed to issue all RACF commands with all operands except the operands that require the AUDITOR attribute. SPECIAL specified on the ADDUSER command overrides NOSPECIAL specified on the CONNECT command.

You must have the SPECIAL attribute to enter the SPECIAL operand.

**NOSPECIAL**

Specifies that the new user is not to have the SPECIAL attribute. NOSPECIAL is the default if you omit both SPECIAL and NOSPECIAL.

### TSO

Specifies that when you define a TSO user to RACF, you can enter any of the following suboperands to specify default TSO logon information for that user. Each suboperand defines information that RACF stores in a field within the TSO segment of the user’s profile.

You can control access to an entire TSO segment or to individual fields within the TSO segment by using field-level access checking. For more information, see [z/OS Security Server RACF Security Administrator’s Guide](https://www.ibm.com/products/zos-security-server-racf-security-administrators-guide).

**ACCTNUM(account-number)**

Specifies the user’s default TSO account number when logging on through the TSO/E logon panel. The account number you specify must be protected by a profile in the ACCTNUM general resource class, and the user must be granted READ access to the profile. Otherwise, the user cannot log on to TSO using the specified account number.

Account numbers can consist of any characters, and can be entered with or without single quotation marks. The following rules apply:

- If parentheses, commas, blanks, or semicolons are to be entered as part of the account number, the character string must be enclosed in single quotation marks. For example, if the account number is (123), you must enter ACCTNUM(' (123) ')
- If a single quotation mark is intended to be part of the account number, use two single quotation marks together for each single quotation mark within the string, and enclose the entire string within single quotation marks.

A user can change an account number, or specify an account number if one has not been specified, using the TSO/E logon panel. RACF checks the user’s authorization to the specified account number. If the user is authorized to use the account number, RACF stores the account number in
the TSO segment of the user’s profile, and TSO/E uses it as a default value the next time the user logs on to TSO/E. Otherwise, RACF denies the use of the account number.

Note: When you define an account number on TSO, you can specify 1–40 characters. When you define a TSO account number to RACF, you can specify only 1–39 characters.

COMMAND(command-issued-at-logon)
Specifies the command to be run during TSO/E logon. TSO/E uses this field to prime the COMMAND field of the logon panel. The command value can contain 1–80 characters and consist of any characters. You can enter the value with or without single quotation marks depending on the following rules:

- If the command value contains parentheses, commas, blanks, or semicolons, enclose the character string in single quotation marks.
- If a single quotation mark is intended to be part of the command value, use two single quotation marks together for each single quotation mark within the string, and enclose the entire string within single quotation marks.

Both uppercase and lowercase characters are accepted and maintained in the case in which they are entered. A user can change the command value, or specify a command if one has not been specified, using the TSO/E logon panel.

DEST(destination-id)
Specifies the default destination to which the system routes dynamically-allocated SYSOUT data sets. The destination-id must be 1–7 alphanumeric characters, beginning with an alphabetic or national character.

HOLDCLASS(hold-class)
Specifies the user's default hold class. The specified value must be 1 alphanumeric character, excluding national characters.

If you specify the TSO operand on the ADDUSER command but do not specify a value for HOLDCLASS, RACF uses a default value consistent with current TSO defaults.

JOBCLASS(job-class)
Specifies the user’s default job class. The specified value must be 1 alphanumeric character, excluding national characters.

If you specify the TSO operand on the ADDUSER command but do not specify a value for JOBCLASS, RACF uses a default value consistent with current TSO defaults.

MAXSIZE(maximum-region-size)
Specifies the maximum region size the user can request at logon. The maximum-region-size is the number of 1024-byte units of virtual storage that TSO can create for the user's private address space. The specified value must be an integer 0–65535 if the database is shared with any MVS systems, or 0–2096128 if the database is not shared with any MVS systems.

If you specify the TSO operand on the ADDUSER command but do not specify a value for MAXSIZE, or specify MAXSIZE(0), RACF uses a default value consistent with current TSO defaults.
ADDUSER

If values are specified for both MAXSIZE and SIZE and SIZE is greater than MAXSIZE, RACF sets SIZE equal to MAXSIZE. If a value is specified for only SIZE or MAXSIZE and SIZE is greater than MAXSIZE, the operand is ignored.

If your installation is sharing a database between MVS and VM, the administrator is allowed to specify the largest possible value. If your installation is sharing a database between MVS systems, and one of the systems sharing the database is an MVS system, the administrator must not specify a value greater than 65535.

MSGCLASS(message-class)
Specifies the user’s default message class. The specified value must be 1 alphanumeric character, excluding national characters.

If you specify the TSO operand on the ADDUSER command but do not specify a value for MSGCLASS, RACF uses a default value consistent with current TSO defaults.

PROC(logon-procedure-name)
Specifies the name of the user’s default logon procedure when logging on through the TSO/E logon panel. The name you specify must be 1–8 alphanumeric characters and begin with an alphabetic character. The name must also be defined as a profile in the TSOPROC general resource class, and the user must be granted READ access to the profile. Otherwise, the user cannot log on to TSO using the specified logon procedure.

A user can change a logon procedure, or specify a logon procedure if one has not been specified, using the TSO/E logon panel. TSO/E checks the user’s authorization to the specified logon procedure. If the user is authorized to use the logon procedure, TSO/E uses it for this session and stores the name of the procedure in the TSO segment of the user’s profile for use as the default value the next time the user logs on to TSO/E. Otherwise, TSO/E denies use of the logon procedure.

SECLABEL(security-label)
Specifies the user’s security label if one was entered on the TSO LOGON panel. On subsequent LOGONS, it appears automatically on the panel.

Note: For more information on the relationship between the TSO security label and the user’s security label, see z/OS Security Server RACF Security Administrator’s Guide.

SIZE(default-region-size)
Specifies the minimum region size if the user does not request a region size at logon. The default region size is the number of 1024-byte units of virtual storage available in the user’s private address space at logon. The specified value must be an integer 0–65535 if the database is shared with any MVS systems, or 0–2096128 if the database is not shared with any MVS systems.

A user can change the minimum region size, or specify the minimum region size if one has not been specified, using the TSO/E logon panel. RACF stores this value in the TSO segment of the user’s profile and TSO/E uses it as a default value the next time the user logs on to TSO/E.

If values are specified for both MAXSIZE and SIZE and SIZE is greater than MAXSIZE, RACF sets SIZE equal to MAXSIZE. If a value is specified for only SIZE or MAXSIZE and SIZE is greater than MAXSIZE, the operand is ignored.
ADDUSER

If your installation is sharing a database between MVS and VM, the administrator is allowed to specify the largest possible value. If your installation is sharing a database between MVS systems, and one of the systems sharing the database is an MVS system, the administrator must not specify a value greater than 65535.

SYS(sysout-class)
Specifies the user’s default SYSOUT class. The specified value must be 1 alphanumeric character, excluding national characters.

If you specify the TSO operand on the ADDUSER command but do not specify a value for SYS, RACF uses a default value consistent with current TSO defaults.

UNIT(unit-name)
Specifies the default name of a device or group of devices that a procedure uses for allocations. The specified value must be 1–8 alphanumeric characters.

USERDATA(user-data)
Specifies optional installation data defined for the user. The specified value must be 4 EBCDIC characters. Valid characters are 0–9 and A–F.

UACC(access-authority)
Specifies the default value for the universal access authority for all new resource profiles the user defines while the user’s default group is the user’s current connect group. The universal access authorities are ALTER, CONTROL, UPDATE, READ, and NONE. (RACF does not accept EXECUTE access authority with the ADDUSER command.) If you omit this operand or specify UACC without an access authority, the default is NONE.

This operand is group-related. If a user is subsequently connected to other groups (with the CONNECT command), the user can have a different default universal access authority in each group. Therefore, if the user specifies a different group at logon time or at batch job execution, the user’s default UACC is the UACC of the specified group, not the UACC of the user’s default group.

WHEN
Specifies the days of the week and the hours in the day when the user is allowed to access the system from a terminal. The day-of-week and time restrictions apply only when a user logs on to the system; that is, RACF does not force the user off the system if the end-time occurs while the user is logged on. Also, the day and time restrictions do not apply to batch jobs; the user can submit a batch job on any day and at any time.

If you omit the WHEN operand, the user can access the system at any time. If you specify the WHEN operand, you can restrict the user’s access to the system to certain days of the week or to a certain time period within each day. Otherwise, you can restrict access to both certain days of the week and to a certain time period within each day.

DAYS(day-info)
Specifies the days of the week when a user may access the system. The day-info value can be any one of the following:

ANYDAY The user can access the system on any day. If you omit DAYS, ANYDAY is the default.

WEEKDAYS The user can access the system only on weekdays (Monday through Friday).

day ... The user can access the system only on the days specified,
ADDUSER

where day can be MONDAY, TUESDAY, WEDNESDAY, THURSDAY, FRIDAY, SATURDAY, or SUNDAY, and you can specify the days in any order.

TIME(time-info)
Specifies the time period each day when the user can access the system. The time-info value can be any one of the following:

ANYTIME Specifies that the user can access the system at any time. If you omit TIME, ANYTIME is the default.

start-time:end-time Specifies that the user can access the system only during the specified time period. The format of both start-time and end-time is hhmm, where hh is the hour in 24-hour notation (00–23) and mm is the minutes (00–59). Note that 0000 is not a valid time value.

If start-time is greater than end-time, the interval spans midnight and extends into the following day.

If you omit DAYS and specify TIME, the time restriction applies to all seven days of the week. If you specify both DAYS and TIME, the user can access the system only during the specified time period and only on the specified days.

WORKATTR
Specifies the user-specific attributes of a unit of work. z/OS elements or features such as APPC, WLM, and z/OS UNIX might use the WORKATTR segment.

WAACNT(account-number)
Specifies an account number for APPC/MVS processing.

You can specify a maximum of 255 EBCDIC characters.

Use the following rules when entering a value for this field:

• If the account number contains parentheses, commas, blanks, or semicolons, enclose the character string in single quotation marks. For example, if the account number is (123), you must enter WAACNT(' (123 ) ').

• If a single quotation mark is intended to be part of the account number, use two single quotation marks together for each single quotation mark within the string, and enclose the entire string within single quotation marks.

WAADDRn(address-line)
Specifies up to four additional address lines for SYSOUT delivery. n can be any number 1–4.

For each address-line, you can specify a maximum of 60 EBCDIC characters. Both uppercase and lowercase characters are accepted and maintained in the case in which they are entered.

Use the following rules when entering a value for this field:

• If the data contains parentheses, commas, blanks, or semicolons, enclose the character string in single quotation marks. For example, if the data is (123), you must enter WAADDR(' (123 ) ').

• If a single quotation mark is intended to be part of the data, use two single quotation marks together for each single quotation mark within the string, and enclose the entire string within single quotation marks.
ADDUSER

WABLDG(*building*)
Specifies the building that SYSOUT information is to be delivered to.

You can specify a maximum of 60 EBCDIC characters. Both uppercase and lowercase characters are accepted and maintained in the case in which they are entered.

Use the following rules when entering a value for this field:
- If the data contains parentheses, commas, blanks, or semicolons, enclose the character string in single quotation marks. For example, if the data is (123), you must enter WABLDG('123').
- If a single quotation mark is intended to be part of the data, use two single quotation marks together for each single quotation mark within the string, and enclose the entire string within single quotation marks.

WADEPT(*department*)
Specifies the department that SYSOUT information is to be delivered to.

You can specify a maximum of 60 EBCDIC characters. Both uppercase and lowercase characters are accepted and maintained in the case in which they are entered.

Use the following rules when entering a value for this field:
- If the data contains parentheses, commas, blanks, or semicolons, enclose the character string in single quotation marks. For example, if the data is (123), you must enter WADEPT('123').
- If a single quotation mark is intended to be part of the data, use two single quotation marks together for each single quotation mark within the string, and enclose the entire string within single quotation marks.

WANAME(*name*)
Specifies the name of the user that SYSOUT information is to be delivered to.

You can specify a maximum of 60 EBCDIC characters. Both uppercase and lowercase characters are accepted and maintained in the case in which they are entered.

Use the following rules when entering a value for this field:
- If the data contains parentheses, commas, blanks, or semicolons, enclose the character string in single quotation marks. For example, if the data is (123), you must enter WANAME('123').
- If a single quotation mark is intended to be part of the data, use two single quotation marks together for each single quotation mark within the string, and enclose the entire string within single quotation marks.

WAROOM(*room*)
Specifies the room that SYSOUT information is to be delivered to.

You can specify a maximum of 60 EBCDIC characters. Both uppercase and lowercase characters are accepted and maintained in the case in which they are entered.

Use the following rules when entering a value for this field:
- If the data contains parentheses, commas, blanks, or semicolons, enclose the character string in single quotation marks. For example, if the data is (123), you must enter WAROOM('123').
ADDUSER

- If a single quotation mark is intended to be part of the data, use two single quotation marks together for each single quotation mark within the string, and enclose the entire string within single quotation marks.

Examples

Table 5. ADDUSER examples

Example 1

**Operation** User IA0 wants to define users PAJ5 and ESH25 to RACF and assign RESEARCH as their default group.

**Known** User IA0 has JOIN authority to group RESEARCH and the CLAUTH attribute for the USER class.

User PAJ5 and ESH25 are not defined to RACF. User IA0 is currently connected to group RESEARCH. User IA0 wants to issue the command as a RACF TSO command.

**Command** ADDUSER (PAJ5 ESH25)

**Defaults** NAME(*************** ) PASSWORD(RESEARCH) OWNER(IA0) DFLTGRP(RESEARCH) AUTHORITY(USE) UACC(NONE) NOGRPACC NOADSP NOSPECIAL NOOPERATIONS NOCLAUTH NOAUDITOR NOOIDCARD

Example 2

**Operation** User WJE10 wants to define user RGH01 to RACF and assign PAYROLL as the default and owning group. The password is PASS, group authority is CREATE, and universal access authority is READ. User WJE10 wants to direct the command to run under the authority of user EPC at ARMK.

**Known** User EPC at ARMK has JOIN authority to group PAYROLL and the CLAUTH attribute for the USER class.

PAYROLL is not the default group of user EPC at ARMK.

User RGH01 is not defined to RACF on node ARMK.

The name of user RGH01 is RG Harris.

User WJE10 wants to issue the command as a RACF TSO command.

User WJE10 and EPC at ARMK have an already established user ID association.

**Command** ADDUSER RGH01 DFLTGRP(PAYROLL) OWNER(PAYROLL) PASSWORD(PASS) NAME('R. G. Harris') AUTHORITY(CREATE) UACC(READ) AT(ARMK.EPC)

**Defaults** NOSPECIAL NOOPERATIONS NOCLAUTH NOOIDCARD NOAUDITOR

Example 3

**Operation** User RACFMIN wants to define user PIZ30 to RACF with a security category of NEWEMPLOYEE and a security level of NOSECRETS. User PIZ30 is to be allowed to use the system only on weekdays between the hours of 8:00 A.M. and 6:00 P.M.

**Known** User RACFMIN has the SPECIAL attribute. NEWEMPLOYEE has been defined to RACF as a valid category, and NOSECRETS has been defined as a valid security level. The new user's name is John Doe. User RACFMIN wants to issue the command as a RACF TSO command.

**Command** ADDUSER PIZ30 NAME('JOHN DOE') ADDCATEGORY(NEWEMPLOYEE) SECLEVEL(NOSECRETS) WHEN(DAYS(WEEKDAYS) TIME(0800:1800))

**Defaults** OWNER(RACFMIN) NOGRPACC NOSPECIAL NOOPERATIONS NOAUDITOR NOADSP AUTHORITY(USE)
Table 5. ADDUSER examples (continued)

Example 4

**Operation**
User TTU01 wants to define user PIZ33 to RACF. User PIZ33 will be the AUDITOR for the installation, and will have class authority to terminals and tape volumes. User PIZ33 will not be required to enter a password, but will be identified through an OIDCARD.

**Known**
User TTU01 has the SPECIAL attribute.

User TTU01 is connected to group RESEARCH.

User PIZ33 is not defined to RACF.

User TTU01 wants to issue the command as a RACF TSO command.

**Command**
Entered in the TSO foreground:
ADDUSER PIZ33 NOPASSWORD OIDCARD CLAUTH(TAPEVOL TERMINAL) AUDITOR

User TTU01 is prompted to enter the OIDCARD for PIZ33.

**Defaults**
NAME(####################) OWNER(TTU01) DFLTGRP(RESEARCH)
AUTHORITY(USE) UACC(NONE) NOGRPACC NOADSP NOSPECIAL NOOPERATIONS

Example 5

**Operation**
User TTU5 wants to define user RADMIN to RACF. User RADMIN will be a member of, and be owned by, the SYSINV group and have a model name of RADMIN.RACF.ACCESS.

**Known**
User TTU5 has at least JOIN authority to group SYSINV and the CLAUTH attribute for the USER class. USER TTU5 wants to issue the command as a RACF TSO command.

**Command**
ADDUSER RADMIN DFLTGRP(SYSINV) MODEL(RACF.ACCESS) NAME('RACF ADMINISTRATOR') AUTHORITY(JOIN) ADSP UACC(NONE) OWNER(SYSINV)

**Defaults**
NOGRPACC, NOSPECIAL, NOOOPERATIONS, NOAUDITOR

Example 6

**Operation**
User KLEWIS wants to define user TBURNS to RACF and assign TSOTEST as the default group and TSOADMN as the owner of the user profile for TBURNS. The user will be allowed to use TSO and will be assigned the following TSO logon information:

- Account number 98765T
- Logon procedure TSPROC3
- Default job class Z
- Default message class Q
- Default hold class X
- SYSOUT class W
- Default region size of 2500
- Maximum region size of 15000.

**Known**
User KLEWIS has the SPECIAL attribute.

- 98765T has been defined to RACF as a profile in the ACCTNUM general resource class, and user TBURNS has been given READ access to this profile.
- TSPROC3 has been defined to RACF as a profile in the TSOPROC general resource class, and user TBURNS has been given READ access to this profile.

- User TBURNS is not defined to RACF.
- User TBURNS's name is T. F. Burns.
- User KLEWIS wants to issue the command as a RACF TSO command.

**Command**
ADDUSER TBURNS DFLTGRP(TSOTEST) OWNER(TSOADMN) NAME('T.F. BURNS')
TSO(ACCTNUM(98765T) PROC(TSPROC3) JOBCLASS(Z) MSGCLASS(Q) HOLDCLASS(X) SYS(W) SIZE(2500) MAXSIZE(15000))

**Defaults**
TSO(NOEST) AUTHORITY(USE) UACC(NONE) NOGRPACC NOADSP NOSPECIAL NOOOPERATIONS NOCLAUTH NOAUDITOR NOOIDCARD
Table 5. ADDUSER examples (continued)

Example 7

**Operation**
User JSMITH wants to define user WJONES to RACF and assign SYS05 as the default group and DFPADMN as the owner of the user profile for WJONES. User WJONES is assigned the following default information to be used by DFP when the user creates a new DFP-managed data set:

- Data class DFP4DATA
- Management class DFP4MGMT
- Storage class DFP4STOR
- Data application identifier DFP4APPL.

**Known**
- User JSMITH has the SPECIAL attribute.
- DFP4MGMT has been defined to RACF as a profile in the MGMTCLAS general resource class, and user WJONES has been given READ access to this profile.
- DFP4STOR has been defined to RACF as a profile in the STORCLAS general resource class, and user WJONES has been given READ access to this profile.
- User WJONES is not defined to RACF.
- User WJONES's name is W. E. Jones.
- User JSMITH wants to issue the command as a RACF TSO command.

**Command**
ADDUSER WJONES DFLTGRP(SYS05) OWNER(DFPADMN) NAME('W.E. JONES')
DFP(DATACLAS(DFP4DATA) MGMTCLAS(DFP4MGMT) STORCLAS(DFP4STOR)
DATAAPPL(DFP4APPL))

**Defaults**
AUTHORITY(USE) UACC(NONE) NOGRPACC NOADSP NOSPECIAL
NOOPERATIONS NOCLAUTH NOAUDITOR NOOIDCARD

Example 8

**Operation**
The system administrator wants to define user DAF0 to RACF with her default group set to RESEARCH, her primary language set to American English (ENU) and her secondary language set to German (DEU).

**Known**
The user's name is D. M. Brown. The profile owner is IBMUSER. The system administrator has the SPECIAL attribute. User DAF0 will have JOIN authority to group RESEARCH. The system administrator wants to issue the command as a RACF TSO command.

**Command**
ADDUSER DAF0 DFLTGRP(RESEARCH) NAME('D. M. BROWN') LANGUAGE(
PRIMARY(ENU) SECONDARY(DEU)) OWNER(IBMUSER) AUTHORITY(JOIN)

**Defaults**
UACC(NONE) NOGRPACC NOADSP NOSPECIAL NOOPERATIONS NOCLAUTH NOAUDITOR NOOIDCARD

Example 9

**Operation**
A user with SPECIAL authority requests the addition of a new z/OS UNIX user.

**Known**
The user profile will be owned by the z/OS UNIX administrator's user ID, SYSADM, and will be a member of the existing group SYSOM which is associated with a GID. The user wants to issue the command as a RACF TSO command.

**Command**
ADDUSER CSMITH DFLTGRP(SYSOM) OWNER(SYSADM) NAME('C.J. SMITH')
OMVS_UID(147483647) HOME(/u/CSMITH) PROGRAM(/bin/sh)

Example 10

**Operation**
A user with SPECIAL authority requests the addition of a new DCE user.

**Known**
The user profile is owned by the system administrator's user ID, SYSADM, and is a member of the existing group SYSOM which is associated with a GID. This DCE user has been assigned a DCE UUID of 004386ea-ebb6-1ec3-bcae-10005ac90feb and a DCE principal name of charlie. This z/OS UNIX DCE user is a principal of the /....elvis.memphis.ibm.com DCE cell. The UUID for the /....elvis.memphis.ibm.com DCE cell is 003456ab-ecb7-7de3-ebda-95531ed63dae.

**Command**
ADDUSER CSMITH DFLTGRP(SYSOM) OWNER(SYSADM) NAME('C.J. SMITH')
OMVS_UID(27) HOME(/u/csmith) PROGRAM(/bin/sh))
DCE_UUID(004386ea-ebb6-1ec3-bcae-10005ac90feb) +
DCENAME(charlie) HOMECCELL(/....elvis.memphis.ibm.com) +
HOMEUUID(003456ab-ecb7-7de3-ebda-95531ed63dae)

**Defaults**
DCE(AUTOLOGIN(NO))
ADDUSER

Table 5. ADDUSER examples (continued)

Example 11

**Operation**
Lotus Notes user RACFADM with SPECIAL or UPDATE authority requests the addition of a new user with Lotus Notes and NDS information.

**Known**
The user profile is owned by RACFADM and belongs to RACFADM's current connect group SYSOM.

**Command**
ADDUSER PCUSER1 LNOTES(SNAME('NEW-GUY 1')) NDS(UNAME(DIRADMIN))

**Defaults**
DFLTGRP(SYSOM) OWNER(RACFADM)

Example 12

**Operation**
User RACFADM with SPECIAL or UPDATE authority requests the addition of a new z/OS UNIX user. The user specifies AUTOUID so that RACF will automatically assign an unused UID to the new user.

**Known**
The user profile is owned by RACFADM and belongs to RACFADM's current connect group SYSOM. The BPX.NEXT.USER profile in the FACILITY class has been set up to allow automatic UID assignment.

**Command**
ADDUSER UNIXUSR OMVS(AUTOUID HOME('/u/unixusr') CPUIMEMAX(5000) ASSIZEMAX(4000000))

**Defaults**
DFLTGRP(SYSOM) OWNER(RACFADM)

Example 13

**Operation**
User RACFADM with SPECIAL or UPDATE authority requests the addition of a new z/OS UNIX superuser.

**Known**
The user profile is owned by RACFADM and belongs to RACFADM's current connect group SYSOM. Shared UIDs are being controlled, and at least one superuser already exists, so SHARED must be specified.

**Command**
ADDUSER SUPERGUY OMVS(UID(0) SHARED HOME('/') PROGRAM('/bin/sh')) NOPASSWORD

**Defaults**
DFLTGRP(SYSOM) OWNER(RACFADM)

Example 14

**Operation**
User RACFADM with SPECIAL authority adds the user ID PUBLIC and assigns it restricted access. User IDs RACFU00 and USER004 are added, but are not assigned any restrictions. In this example, the PUBLIC user ID does not have access to RACFU00's data sets because it has RESTRICTED access.

**Known**
User RACFADM has SPECIAL authority.

**Command**
ADDUSER PUBLIC RESTRICTED
ADDUSER RACFU00 NORESTRICTED
ADDUSER USER004
ADDS 'RACFU00.*' UACC(READ)

**Defaults**
USER004 has NORESTRICTED access by default.

Example 15

**Operation**
A user with SPECIAL authority requests the addition of a z/OS Integrated Security Services Network Authentication Service account within the local realm for a user whose RACF user profile is RONTOMS. MAXTTLFE is not specified, so the value specified on the definition of the local realm KERBDFLT in the REALM class is used. Note that the user's RACF password must be changed before the definition of the z/OS Network Authentication Service account is complete.

**Known**
User RONTOMS wants to define his z/OS Integrated Security Services Network Authentication Service information.

**Command**
ADDUSER RONTOMS KERB(KERBNAME('KerberizedUser'))

Example 16

**Operation**
User RACFADNM issues a command to add a new user MRSERVER with an EIM segment and LDAP profile that is related to an LDAPBIND class for the specified user to use with EIM.

**Known**
eimdomainALookup is a profile in the LDAPBIND class that defines the EIM LDAP values required for EIM processing.

**Command**
ADDUSER MRSERVER EIM(LDAPPROF(eimdomainALookup))
Table 5. ADDUSER examples (continued)

**Example 17**

**Operation**  User SECADM wants to define a new user ANDREW and add custom field data for multiple fields.

**Known**  User SECADM has the SPECIAL attribute. Custom fields called EMPSER, ADDRESS, PHONE, CODE, and ACTIVE are already defined with attributes that allow the custom data values specified in the command example. The systems programer has already rebuilt the dynamic parse table using the IRRDPI00 UPDATE command.

**Command**  
ADDUSER ANDREW CSDATA(EMPSER(256400)
  ADDRESS('14 Main Street, Anywhere, IL 01234')
  PHONE(555-555-5555)
  CODE(FC01B2D8)
  ACTIVE(NO))
ALTDSD (Alter data set profile)

Purpose

Use the ALTDSD command to:

- Modify an existing discrete or generic data set profile.
- Protect a single volume of either a multivolume tape data set or a multivolume, non-VSAM DASD data set. (At least one volume must already be RACF-protected.)
- Remove RACF-protection from either a single volume of a multivolume tape data set or a single volume of a multivolume, non-VSAM DASD data set. (You cannot delete the last volume from the profile.)

Changes made to discrete profiles take effect after the ALTDSD command is processed. Changes made to generic profiles do not take effect until one or more of the following steps is taken:

- The user of the data set issues the LISTDSD command:
  LISTDSD DA(data-set-protected-by-the-profile) GENERIC

  **Note**: Use the data set name, not the profile name.

- The security administrator issues the SETROPTS command:
  SETROPTS GENERIC(DATASET) REFRESH
  See SETROPTS command for authorization requirements.

- The user of the data set logs off and logs on again.

  **Note**: For more information, refer to [z/OS Security Server RACF Security Administrator's Guide](#).

Issuing options

The following table identifies the eligible options for issuing the ALTDSD command:

<table>
<thead>
<tr>
<th>As a RACF TSO command?</th>
<th>As a RACF operator command?</th>
<th>With command direction?</th>
<th>With automatic command direction?</th>
<th>From the RACF parameter library?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

For information on issuing this command as a RACF TSO command, refer to Chapter 3, “RACF TSO commands,” on page 15.

For information on issuing this command as a RACF operator command, refer to Chapter 4, “RACF operator commands,” on page 21.

You must be logged on to the console to issue this command as a RACF operator command.

Related commands

- To create a data set profile, see [ADDSD (Add data set profile)](#).
- To delete a data set profile, see [DELDSD (Delete data set profile)](#).
- To list a data set profile, see [LISTDSD (List data set profile)](#).
- To permit or deny access to a data set profile, see [PERMIT (Maintain resource access lists)](#).
ALTDSD

- To obtain a list of data set profiles, see SEARCH (Search RACF database).

**Authorization required**

When issuing the ALTDSD command as a RACF operator command, you require sufficient authority to the proper resource in the OPERCMDS class. See z/OS Security Server RACF Security Administrator's Guide for further information.

To use the ALTDSD command, you must have sufficient authority over the profile. RACF makes the following checks until one of these conditions is met:

- You have the SPECIAL attribute.
- The data set profile is within the scope of a group in which you have the group-SPECIAL attribute.
- You are the owner of the profile.
- The high-level qualifier of the profile name (or the qualifier supplied by the RACF naming conventions table or by a command installation exit) is your user ID.
- To assign a security label, you must have the SPECIAL attribute or have READ access to the security label profile. However, the security administrator can limit the ability to assign security labels only to users with the SPECIAL attribute.
- To access the DFP or TME segment, field-level access checking is required.
- For a discrete profile, you are in the access list for the discrete profile and you have ALTER authority. (If you have any other level of authority, you cannot alter this profile.)
- For a discrete profile, your current connect group (or, if list-of-groups checking is active, any group to which you are connected) is in the access list and has ALTER authority.
- For a discrete profile, the universal access authority is ALTER.

To use the GLOBALAUDIT operand, you must have the AUDITOR attribute, or the data set profile must be within the scope of a group in which you have the group-AUDITOR attribute.

If you have the AUDITOR attribute or the data set profile is within the scope of a group in which you have the group-AUDITOR attribute, but you do not satisfy one of the above checks, you can specify only the GLOBALAUDIT operand.

To specify the AT keyword, you must have READ authority to the DIRECT.node resource in the RRSFDATA class and a user ID association must be established between the specified node.userid pair(s).

To specify the ONLYAT keyword you must have the SPECIAL attribute, the userid specified on the ONLYAT keyword must have the SPECIAL attribute, and a user ID association must be established between the specified node.userid pair(s) if the user IDs are not identical.

To assign a security category to a profile, you must have the SPECIAL attribute, or the access category must be in your user profile. To assign a security level to a profile, you must have the SPECIAL attribute, or, in your own profile, a security level that is equal to, or greater than, the security level you are assigning.

**Syntax**

For the key to the symbols used in the command syntax diagrams, see Syntax of RACF commands and operands on page 9. The complete syntax of the ALTDSD command is:
ALTDSD

[subsystem-prefix][ALTDSD | ALD]

(profile-name [ /password ] . . .)

[ ADDCATEGORY(category-name ...)
  | DELCATEGORY ([ (category-name ... | * ) ] ) ]
[ ADDVOL(volume-serial)
  | DELVOL(volume-serial)
  | ALTVOL(old-volume-serial new-volume-serial) ]
[ AT([node].userid ...) | ONLYAT([node].userid ...) ]
[ AUDIT(access-attempt[(audit-access-level)] ...) ]
[ DATA('installation-defined-data') | NODATA ]
[ DFP(RESOWNER(userid or group-name) | NORESOWNER)
  | NODFP ]
[ ERASE | NOERASE ]
[ GENERIC | SET | NOSET ]
[ GLOBALAUDIT(access-attempt[(audit-access-level)] ...) ]
[ LEVEL(nn) ]
[ NOTIFY(userid) | NONOTIFY ]
[ OWNER(userid or group-name) ]
[ RETPD(nnnnn) ]
[ SECLABEL(seclabel-name) | NOSECLABEL ]
[ SECLEVEL(seclvel-name) | NOSECLEVEL ]
[ TME(
  [ ROLES(role-access-specification ...)
  | ADDROLES(role-access-specification ...)
  | DELROLES(role-access-specification ...)
  | NOROLES ]
  )
  | NOTME ]
[ UACC(access-authority) ]
[ UNIT(type) ]
[ VOLUME(volume-serial) ]
[ WARNING | NOWARNING ]

For information on issuing this command as a RACF TSO command, refer to Chapter 3, "RACF TSO commands," on page 15.

For information on issuing this command as a RACF operator command, refer to Chapter 4, "RACF operator commands," on page 21.

Parameters

subsystem-prefix
Specify that the RACF subsystem is the processing environment of the command. The subsystem prefix can be either the installation-defined prefix for RACF (1–8 characters) or, if no prefix has been defined, the RACF subsystem name followed by a blank. If the command prefix was registered with CPF, you can use the MVS command D OPDATA to display it or you can contact your RACF security administrator.

Only specify the subsystem prefix when issuing this command as a RACF operator command. The subsystem prefix is required when issuing RACF operator commands.

profile-name
Specify the name of a discrete or generic data set profile. If you specify more than one profile name, the list of names must be enclosed in parentheses.
This operand is required and must be the first operand following ALTDSD.

Notes:
1. Because RACF uses the RACF database and not the system catalog, you cannot use alias data set names.
2. If you specify a generic profile name, RACF ignores these operands:
   - ADDVOL | DELVOL | ALTVOL
   - SET | NOSET
   - UNIT
   - VOLUME

/password
Specifies the data set password if you are altering the profile for a password-protected data set. This operand applies only if you are using the ADDVOL and SET operands for a volume of a multivolume password-protected data set. The WRITE level password must then be specified.

If the command is executing in the foreground and you omit the password for a password-protected data set, RACF uses the logon password. You are prompted if the password you enter or the logon password is incorrect.

If the command is executing in a batch job and you either omit the password for a password-protected data set or supply an incorrect password, the operator is prompted.

You can use this operand only for tape data sets and non-VSAM DASD data sets. If you specify a generic profile, RACF ignores this operand.

ADDCATEGORY | DELCATEGORY

ADDCATEGORY(category-name ...)
Specifies one or more names of installation-defined security categories. category-name must be defined as a member of the CATEGORY profile in the SECDATA class. (For information on defining security categories, see z/OS Security Server RACF Security Administrator’s Guide.)

Specifying ADDCATEGORY on the ALTDSD command causes RACF to add any category names you specify to any list of required categories that already exists in the data set profile. All users previously allowed to access the data set can continue to do so only if their profiles also include the additional category names.

When the SECDATA class is active and you specify ADDCATEGORY, RACF performs security category checking in addition to its other authorization checking. If a user requests access to a data set, RACF compares the list of security categories in the user profile with the list of security categories in the data set profile. If RACF finds any security category in the data set profile that is not in the user’s profile, RACF denies access to the data set. If the user’s profile contains all the required security categories, RACF continues with other authorization checking.

Note: RACF does not perform security category checking for a started task or user that has the RACF trusted or privileged attribute. The RACF trusted or privileged attribute can be assigned to a started task through the RACF started procedures table or STARTED class, or to other users by installation-supplied RACF exits.
**DELCATEGORY**

Specifies one or more names of installation-defined security categories you want to delete from the data set profile. Specifying an asterisk (*) deletes all categories; RACF no longer performs security category checking for the data set profile.

Specifying DELCATEGORY by itself causes RACF to delete from the profile only undefined category names (those category names that were once known to RACF but that the installation has since deleted from the CATEGORY profile.)

**ADDVOL | DELVOL | ALTVOL**

**ADDVOL**(volume-serial)

Specifies that you want to RACF-protect the portion of the data set residing on this volume. At least one other portion of the data set on a different volume must already have been RACF-protected. You can use this operand only for tape data sets and non-VSAM data sets.

The DASD volume must be online unless you also specify NOSET. If it is not online and you omit NOSET, the ALTDSD command processor will, if you have TSO MOUNT authority, request that the volume be mounted.

RACF ignores this operand if you specify a generic profile name.

**Note:** The maximum number of volume serials for a tape data set with an entry in the TVTOC is 42.

**DELVOL**(volume-serial)

Specifies that you want to remove RACF-protection from the portion of the data set residing on this volume. If no other portions of this data set on another volume are RACF-protected, the command terminates. (Use the DELDSD command to delete the profile from RACF.) You can use this operand only for tape data sets and non-VSAM DASD data sets.

The DASD volume must be online unless you also specify NOSET. If it is not online and you omit NOSET, the ALTDSD command processor requests that the volume be mounted.

RACF ignores this operand if you specify a generic profile name.

**ALTVOL**(old-volume-serial new-volume-serial)

Specifies that you want to change the volume serial number in the data set profile. You can specify this operand for both VSAM and non-VSAM DASD data sets, but you cannot specify it for tape data sets. If you specify ALTVOL for a tape data set, the command fails.

When you specify ALTVOL, RACF ignores the SET and NOSET operands and modifies the data set profile, but it does not process the RACF indicator.

RACF ignores this operand if you specify a generic profile name.

To specify ALTVOL, you must have the SPECIAL attribute, or the data set profile must be within the scope of a group in which you have the group-SPECIAL attribute, or the high-level qualifier of the data set name (or the qualifier supplied by a command installation exit routine) must be your user ID.

**AT | ONLYAT**

The AT and ONLYAT keywords are only valid when the command is issued as a RACF TSO command.
AT([node].userid ...) Specifies that the command is to be directed to the node specified by node, where it runs under the authority of the user specified by userid in the RACF subsystem address space.

If node is not specified, the command is directed to the local node.

ONLYAT([node].userid ...) Specifies that the command is to be directed only to the node specified by node where it runs under the authority of the user specified by userid in the RACF subsystem address space.

If node is not specified, the command is directed only to the local node.

AUDIT(access-attempt[(audit-access-level)] ...) Specifies which access attempts and access levels the user who has the AUDITOR attribute wants logged to the SMF data set.

access-attempt Specifies which new access attempts you want logged to the SMF data set. The following options are available:

- ALL Specifies that you want to log both authorized accesses and detected unauthorized access attempts.
- FAILURES Specifies that you want to log detected unauthorized access attempts.
- NONE Specifies that you do not want any logging to be done.
- SUCCESS Specifies that you want to log authorized accesses.

If you specify AUDIT without a value, RACF ignores it.

audit-access-level Specifies which access levels you want logged to the SMF data set. The levels you can specify are:

- ALTER Logs ALTER access-level attempts only.
- CONTROL Logs access attempts at the CONTROL and ALTER levels.
- READ Logs access attempts at any level. READ is the default value if you omit audit-access-level.
- UPDATE Logs access attempts at the UPDATE, CONTROL, and ALTER levels.

You cannot audit access attempts at the EXECUTE level.

DATA | NODATA

DATA('installation-defined-data') Specifies up to 255 characters of installation-defined data to be stored in the data set profile and must be enclosed in single quotation marks. It can also contain double-byte character set (DBCS) data.

Use the LISTDSD command to list this information.

NODATA Specifies that the ALTDSD command is to delete any installation-defined data in the data set profile.

DFP | NODFP
DFP
Specifies that for an SMS-managed data set, you can change the following information:

RESOWNER(userid or group-name) I NORESOWNER
Specifies the user ID or group name of the actual owner of the data sets protected by the profile specified in profile-name-1. The name specified for RESOWNER must be a RACF-defined user or group. (The data set resource owner, or RESOWNER, is distinguished from the OWNER, which represents the user or group that owns the data set profile).

If NORESOWNER is specified, the user or group represented by the high level qualifier of the data set profile is assigned as the owner of data sets protected by the profile when SMS needs to determine the RESOWNER.

You can control access to the entire DFP segment or to individual fields within the DFP segment by using field-level access checking. For more information, see z/OS Security Server RACF Security Administrator's Guide.

NODFP
Specifies that RACF should delete the DFP segment from the data set profile.

ERASE | NOERASE

ERASE
Specifies that when SETROPTS ERASE is active, data management is to physically erase the contents of deleted data sets and scratched or released DASD extents. Erasing the data set means overwriting its contents with binary zeroes so that it cannot be read.

Restrictions: The ERASE operand is ignored when any of the following conditions exist:

- When the data set is a tape data set and your installation did not activate the TAPEAUTHDSN option in the DEVSUPxx member of SYS1.PARMLIB. See Erasing Scratched or Release Data (ERASE Option) in z/OS Security Server RACF Security Administrator's Guide for more information.
- When SETROPTS NOERASE is active for your installation. (User and data set profile definitions are overridden.)

NOERASE
Specifies that data management is not to physically erase the contents of deleted data sets and scratched or released DASD extents.

Restrictions: Setting NOERASE has no effect and does not prevent a scratched data set from being erased for either one of the following conditions:

- SETROPTS ERASE(ALL) is in effect.
- SETROPTS ERASE(SECLEVEL(security-level)) is in effect and the scratched data set has security level that is equal or higher than the security-level specified with SETROPTS.

GENERIC | SET | NOSET
If you do not specify GENERIC, SET, or NOSET, the default value is SET.
GLOBALAUDIT(access-attempt[(audit-access-level)] ...)

Specifies which access attempts and access levels the user who has the AUDITOR attribute wants logged to the SMF data set.

access-attempt

Specifies which access attempts the user who has the AUDITOR attribute wants logged to the SMF data set. The following options are available:

ALL

Specifies that you want to log both authorized accesses and detected unauthorized access attempts.

FAILURES

Specifies that you want to log detected unauthorized access attempts.

NONE

Specifies that you do not want any logging to be done.

SUCCESS

Specifies that you want to log authorized accesses.

If you specify AUDIT without a value, RACF ignores it.

audit-access-level

Specifies which access levels the user who has the AUDITOR attribute wants logged to the SMF data set. The levels you can specify are:

ALTER

Logs ALTER access-level attempts only.

CONTROL

Logs access attempts at the CONTROL and ALTER levels.
ALTDSD

**READ**
Logs access attempts at any level. READ is the default value if you omit `audit-access-level`.

**UPDATE**
Logs access attempts at the UPDATE, CONTROL, and ALTER levels.

You cannot audit access attempts at the EXECUTE level.

To use the GLOBALAUDIT operand, you must have the AUDITOR attribute, or the profile must be within the scope of a group in which you have the group-AUDITOR attribute.

**Note:** Regardless of the value specified in GLOBALAUDIT, RACF always logs all access attempts specified on the AUDIT operand.

**LEVEL**(nn)
Specifies a new level indicator, where `nn` is an integer 0–99.

Your installation assigns the meaning of the value.

RACF includes it in all records that log data set accesses and in the LISTDSD command display.

**NOTIFY** | **NONOTIFY**

**NOTIFY**(`userid`)
Specifies the user ID of a user to be notified whenever RACF uses this profile to deny access to a data set. If you specify NOTIFY without specifying a user ID, RACF takes your user ID as the default; you are notified whenever the profile denies access to a data set.

A user who is to receive NOTIFY messages should log on frequently, both to take action in response to the unauthorized access attempts the messages describe and to clear the messages from the SYS1.BROADCAST data set. (When the profile also includes WARNING, RACF might have granted access to the data set to the user identified in the message.)

**Note:** The user ID specified on the NOTIFY operand is not notified when the profile disallows creation or deletion of a data set. NOTIFY is only used for resource access checking, not for resource creation or deletion.

**NONOTIFY**
Specifies that no user is to be notified when RACF uses this profile to deny access to a data set.

**OWNER**(userid or group-name)
Specifies a RACF-defined user or group to be the new owner of the data set profile. If you specify a user ID as the owner of a group data set profile, the specified user must have at least USE authority in the group to which the data set profile belongs.

To change the owner of a profile, you must be the current owner of the profile or have the SPECIAL attribute, or the profile must be within the scope of a group in which you have the group-SPECIAL attribute.

**Note:** The user specified as the owner does not automatically have access to the data set. Use the PERMIT command to add the owner to the access list as desired.
RETPD(nn.nn)
Specifies the RACF security retention period for a tape data set. The security
retention period is the number of days that must elapse before a tape data set
profile expires. (Note that, even though the data set profile expires,
RACF-protection for data sets protected by the profile is still in effect. For more
information, see [z/OS Security Server RACF Security Administrator's Guide](#) )

The number you specify must be 1 to 5 digits in the range of 0 through 65533
or, to indicate a data set that never expires, 99999.

Using RETPD to change the RACF security retention period for a data set
means that the RACF security retention period and the data set retention period
specified by the EXPDT/RETPD parameters on the JCL DD statement are
longer be the same.

When the TAPEVOL class is active, RACF checks the RACF security retention
period before it allows a data set to be overwritten. RACF adds the number of
days in the retention period to the creation date for the data set. If the result is
less than the current date, RACF continues to protect the data set.

When the TAPEVOL class is not active, RACF ignores the RETPD operand.

Specifying this operand for a DASD data set does not cause an error, but it has
no meaning because RACF ignores the operand during authorization checking.

SECLABEL | NOSECLABEL

SECLABEL(seclabel-name)
Specifies an installation-defined security label for this profile. A security
label corresponds to a particular security level (such as CONFIDENTIAL)
with a set of zero or more security categories (such as PAYROLL or
PERSONNEL).

RACF stores the name of the security label you specify in the data set
profile if you are authorized to use that SECLABEL.

If you are not authorized to the SECLABEL or if the name you had specified
is not defined as a SECLABEL profile in the SECLABEL class, the data set
profile is not updated.

Note: If the SECLABEL class is active and the security label is specified in
this profile, any security levels and categories in the profile are
ignored.

NOSECLABEL
removes the security label, if one had been specified, from the profile.

SECLEVEL | NOSECLEVEL

SECLEVEL(secelvel-name)
Specifies the name of an installation-defined security level. This name
 corresponds to the number that is the minimum security level that a user
 must have to access the data set. The secelvel-name must be a member of
 the SECLEVEL profile in the SECDATA class.

When you specify SECLEVEL and the SECDATA class is active, RACF
 adds security level access checking to its other authorization checking. If
global access checking does not grant access, RACF compares the
security level allowed in the user profile with the security level required in
the data set profile. If the security level in the user profile is less than the
security level in the data set profile, RACF denies the access. If the security
level in the user profile is equal to or greater than the security level in the
data set profile, RACF continues with other authorization checking.

**Note:** RACF does not perform security level checking for a started task or
user that has the RACF privileged or trusted attribute. The RACF
privileged or trusted attribute can be assigned to a started task
through the RACF started procedures table or STARTED class, or to
other users by installation-supplied RACF exits.

If the SECDATA class is not active, RACF stores the name you specify in
the data set profile. When the SECDATA class is activated and the name
you specified is defined as a SECLEVEL profile, RACF can perform security
level access checking for the data set profile. If the name you specify is not
defined as a SECLEVEL profile and the SECDATA class is active, you are
prompted to provide a valid security level name.

**NOSECLEVEL**
Specifies that the ALTDSD command is to delete the security level name
from the profile. RACF no longer performs security level access checking
for the data set.

**TME I NOTME**

**TME**
Specifies that information for the Tivoli Security Management Application is
to be added, changed, or deleted.

**Note:** The TME segment fields are intended to be updated only by the
Tivoli Security Management Application, which manages updates,
permissions, and cross references. A security administrator should
only directly update Tivoli Security Management fields on an
exception basis.

**ROLES(role-access-specification ...)**
Specifies a list of roles and associated access levels related to this
profile.

One or more role-access-specification values can be specified, each
separated by blanks. Each value should contain no imbedded blanks
and should have the following format:

```
role-name:authority[;conditional-class:conditional-profile]
```

where role-name is a discrete general resource profile defined in the
ROLE class. The authority is the access authority (NONE, EXECUTE,
READ, UPDATE, CONTROL, or ALTER) with which groups in the role
definition should be permitted to the resource.

The conditional-class is a class name (APPCPORT, CONSOLE,
JESINPUT, PROGRAM, TERMINAL, or SYSID) for conditional access
permission, and is followed by the conditional-profile value, a resource
profile defined in the conditional class.

**ADDROLES(role-access-specification ...)**
Specifies that specific roles and access levels are to be added to the
current list.

**DELROLES(role-access-specification ...)**
Specifies that specific roles from the current list of roles are to be
removed.
ALTDSD

NOROLES
   Specifies that the entire list of roles be removed.

NOTME
   Specifies that RACF delete the TME segment from the profile.

UACC(access-authority)
   Specifies the universal access authority to be associated with the data sets.
   The universal access authorities are ALTER, CONTROL, READ, UPDATE, EXECUTE, and NONE. If you specify CONTROL for a tape data set or a non-VSAM DASD data set, RACF treats the access authority as UPDATE. If you specify EXECUTE for a tape data set or a DASD data set not used as a program library, RACF treats the access authority as NONE.
   If a user accessing a data set has the RESTRICTED attribute, RACF treats the universal access authority (UACC) as NONE for that access attempt.
   If you enter UACC without a value, RACF retains the old universal access authority for the data sets.

UNIT(type)
   Specifies the unit type to be added to the data set profile on which a non-VSAM data set resides. You can specify an installation-defined unit name, a generic device type, or a specific device address. RACF ignores this operand if you specify a generic profile name.

VOLUME(volume-serial)
   Specifies the volume on which the tape data set, the non-VSAM DASD data set, or the catalog for the VSAM data set resides.
   If you specify VOLUME and volume-serial does not appear in the profile for the data set, the command fails. If you omit VOLUME and the data set name appears more than once in the RACF database, the command fails. If you omit VOLUME and the data set name appears only once in the RACF database, no volume serial checking is performed and processing continues.
   RACF ignores this operand if you specify a generic profile name.

WARNING | NOWARNING

WARNING
   Specifies that even if access authority is insufficient, RACF is to issue a warning message and allow access to the resource. RACF also records the access attempt in the SMF record if logging is specified in the profile.
   When SETROPTS MLACTIVE(FAILURES) is in effect: A user or task can access a data set that is in WARNING mode and has no security label even when MLACTIVE(FAILURES) is in effect and the class requires security labels. The user or task receives a warning message and gains access.

NOWARNING
   Specifies that if access authority is insufficient, RACF is to deny the user access to the resource and not issue a warning message.
## ALTDSD

### Examples

#### Table 6. ALTDSD examples

| Example 1 | Operation | User AEH0 owns data set profile PAYROLL.DEPT2.DATA and wants to assign ownership of the data set to group PAYROLL. Only users with categories of FINANCIAL and PERSONNEL and a security level of PERSONAL are to be able to access the data set.  
Known | Data set PAYROLL.DEPT2.DATA is RACF-defined with a discrete profile. FINANCIAL and PERSONNEL are valid categories of access; PERSONAL is a valid security level name. USER AEH0 wants to issue the command as a RACF TSO command.  
Command | `ALTDSD 'PAYROLL.DEPT2.DATA' OWNER(PAYROLL) ADDCATEGORY(FINANCIAL PERSONNEL) SECLEVEL(PERSONAL)`  
Defaults | None. |
|---|---|---|
| Example 2 | Operation | User WRH0 wants to change the universal access authority to NONE for data set RESEARCH.PROJ02.DATA and wants to have all accesses to the data set logged on SMF records. User ADMIN02 is to be notified when RACF uses this profile to deny access to the data set. The data set is to be erased when it is deleted (scratched).  
Known | User WRH0 has ALTER access to data set profile RESEARCH.PROJ02.DATA. User WRH0 is logged onto group RESEARCH. USER WRH0 wants to issue the command as a RACF TSO command.  
User ADMIN02 is a RACF-defined user.  
Data set RESEARCH.PROJ02.DATA is RACF-defined with a generic profile. The SETROPTS ERASE option has been specified for the installation.  
Command | `ALTDSD 'RESEARCH.PROJ02.DATA' UACC(NONE) AUDIT(ALL(READ)) GENERIC NOTIFY(ADMIN02) ERASE`  
Defaults | None. |
| Example 3 | Operation | User CD0 wants to remove RACF-protection from volume 222222 of the multivolume data set CD0.PROJ2.DATA.  
Known | CD0.PROJ2.DATA is a non-VSAM data set that resides on volumes 111111 and 222222 and is defined to RACF with a discrete profile. Volume 222222 is online. User CD0's TSO profile specifies PREFIX (CD0). User CD0 wants to issue the command as a RACF operator command, and the RACF subsystem prefix is @.  
Command | `@ALTDSD PROJ2.DATA DELVOL(222222)`  
Defaults | None. |
| Example 4 | Operation | User RVD02 wants to have all successful accesses to data set PAYROLL.ACCOUNT on volume SYS003 to be logged to the SMF data set.  
Known | User RVD02 has the AUDITOR attribute. User RVD02 wants to issue the command as a RACF TSO command.  
Command | `ALTDSD 'PAYROLL.ACCOUNT' GLOBALAUDIT(SUCCESS(READ)) VOLUME(SYS003)`  
Defaults | None. |
| Example 5 | Operation | User SJR1 wants to modify the installation-defined information associated with the tape data set SYSINV.ADMIN.DATA. The RACF security retention period is to be 360 days.  
Known | User SJR1 has ALTER authority to the data set profile. User SJR1 wants to issue the command as a RACF TSO command.  
Data set protection is active.  
Command | `ALTDSD 'SYSINV.ADMIN.DATA' DATA('LIST OF REVOKED RACF USERIDS') RETPD(360)`  
Defaults | None. |
Table 6. ALTDSD examples (continued)

**Example 6**

*Operation* User ADM1 wants to log all unauthorized access attempts and all successful updates to data sets protected by a generic profile (SALES.ABC.*).

*Known* User ADM1 has the SPECIAL attribute. User ADM1 wants to issue the command as a RACF TSO command.

*Command* `ALTDSD 'SALES.ABC.*' AUDIT (FAILURES(READ) SUCCESS (UPDATE))`

*Defaults* None.

**Example 7**

*Operation* User ADM1 owns the DFP-managed data set RESEARCH.TEST.DATA3 and wants to assign user ADM6 as the data set resource owner.

User ADM1 wants to direct the command to run at node CLCON under the authority of user DROLLO and prohibit the command from being automatically directed to other nodes.

*Known* Data set RESEARCH.TEST.DATA3 is RACF-defined with a discrete profile. Users ADM1 and DROLLO at CLCON have the SPECIAL attribute, and ADM6 is defined to RACF on node CLCON. User ADM1 wants to issue the command as a RACF TSO command. Users ADM1 and DROLLO at CLCON have an already established user ID association.

*Command* `ALTDSD 'RESEARCH.TEST.DATA3' DFP(RESOWNER(ADM6)) ONLYAT(CLCON.DROLLO)`

*Results* The command is only processed on the node CLCON and not automatically directed to any other nodes in the RRSF configuration.
ALTGROUP

ALTGROUP (Alter group profile)

Purpose

Use the ALTGROUP command to change:
- The superior group of a group
- The owner of a group
- The terminal indicator for a group
- A model profile name for a group
- The installation-defined data associated with a group
- The default segment information for a group (for example, DFP or OMVS)

Issuing options

The following table identifies the eligible options for issuing the ALTGROUP command:

<table>
<thead>
<tr>
<th>As a RACF TSO command?</th>
<th>As a RACF operator command?</th>
<th>With command direction?</th>
<th>With automatic command direction?</th>
<th>From the RACF parameter library?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

For information on issuing this command as a RACF TSO command, refer to Chapter 3, “RACF TSO commands,” on page 15.

For information on issuing this command as a RACF operator command, refer to Chapter 4, “RACF operator commands,” on page 21.

You must be logged on to the console to issue this command as a RACF operator command.

Related commands

- To create a group profile, see ADDGROUP (Add group profile)
- To delete a group profile, see DELGROUP (Delete group profile)
- To connect a user to a group, see CONNECT (Connect user to group)
- To list information for a group profile, see LISTGRP (List group profile)
- To remove a user from a group, see REMOVE (Remove user from group)
- To obtain a list of group profiles, see SEARCH (Search RACF database)

Authorization required

When issuing the ALTGROUP command as a RACF operator command, you might require sufficient authority to the proper resource in the OPERCMDS class. See z/OS Security Server RACF Security Administrator’s Guide for further information.

To change the superior group of a group, you must have at least one of the following authorizations:
- You must have the SPECIAL attribute
- All the following group profiles must be within the scope of a group in which you have the group-SPECIAL attribute:
  - The group whose superior group you are changing
  - The current superior group
ALTGROUP

- The new superior group
- You must be the owner of, or have JOIN authority in, both the current and the
  new superior groups.

**Note:** You can have JOIN authority in one group and be the owner of or have the
  group-SPECIAL attribute in the other group.

If you have any of the following authorizations, you can specify any operand except
  as otherwise listed below:
- The SPECIAL attribute
- The group profile is within the scope of a group in which you have the
  group-SPECIAL attribute
- You are the current owner of the group.

To add, delete, or alter segments, such as DFP or OMVS, in a group’s profile, you
  must have at least one of the following authorizations:
- You must have the SPECIAL attribute.
- Your installation must permit you to do so through field-level access checking.

For information on field-level access checking, see [z/OS Security Server RACF
  Security Administrator’s Guide](#).

To specify the AT keyword, you must have READ authority to the DIRECT.node
  resource in the RRSFDATA class and a user ID association must be established
  between the specified node.userid pair(s).

To specify the ONLYAT keyword you must have the SPECIAL attribute, the userid
  specified on the ONLYAT keyword must have the SPECIAL attribute, and a user ID
  association must be established between the specified node.userid pair(s) if the
  user IDs are not identical.

To specify the SHARED keyword, you must have the SPECIAL attribute or at least
  READ authority to the SHARED.IDS resource in the UNIXPRIV class.

**Syntax**

For the key to the symbols used in the command syntax diagrams, see [Syntax of
  RACF commands and operands](#) on page 9. The complete syntax of the
  ALTGROUP command is:
Parameters

subsystem-prefix

Specifies that the RACF subsystem is the processing environment of the command. The subsystem prefix can be either the installation-defined prefix for RACF (1–8 characters) or, if no prefix has been defined, the RACF subsystem name followed by a blank. If the command prefix was registered with CPF, you can use the MVS command D OPDATA to display it or you can contact your RACF security administrator.

Only specify the subsystem prefix when issuing this command as a RACF operator command. The subsystem prefix is required when issuing RACF operator commands.

For information on issuing this command as a RACF TSO command, refer to Chapter 3, “RACF TSO commands,” on page 15.

For information on issuing this command as a RACF operator command, refer to Chapter 4, “RACF operator commands,” on page 21.
**ALTGROUP**

*group-name*
Specifies the name of the group whose definition you want to change. If you specify more than one group name, the list of names must be enclosed in parentheses.

This operand is required and must be the first operand following ALTGROUP.

**AT | ONLYAT**
The AT and ONLYAT keywords are only valid when the command is issued as a RACF TSO command.

**AT([node].userid ...)**
Specifies that the command is to be directed to the node specified by `node`, where it runs under the authority of the user specified by `userid` in the RACF subsystem address space.

If `node` is not specified, the command is directed to the local node.

**ONLYAT([node].userid ...)**
Specifies that the command is to be directed only to the node specified by `node` where it runs under the authority of the user specified by `userid` in the RACF subsystem address space.

If `node` is not specified, the command is directed only to the local node.

**CSDATA | NOCSDATA**

**CSDATA**
 Specifies information to add, change, or remove a custom field for this group.

*custom-field-name ... | NOcustom-field-name ...*
Specifies the name and value of a custom field for this group.

You can define multiple custom field values with a single ALTGROUP command.

**custom-field-name(custom-field-value) ...**
Specifies the name and value of a custom field for this group. You can specify values for multiple custom fields with a single ALTGROUP command.

Usage for each custom field is defined using the CFDEF operand of the RDEFINE command for resource profiles in the CFIELD class. Contact your security administrator to see how custom fields are used at your installation. For more information about custom fields, see [z/OS Security Server RACF Security Administrator’s Guide](#).

**Rules:**
- You must use the same `custom-field-name` as defined by the CFIELD profile named GROUP.CSDATA.custom-field-name. (The CFIELD profile is defined using the CFDEF operand of the RDEFINE command.)
- You must specify a `custom-field-value` that is valid for the attributes of this custom field. (The attributes, such as data type, are defined in the CFDEF segment of the CFIELD profile.)

**NOcustom-field-name ...**
Removes the custom field information for this group. You can remove values for multiple custom fields with a single ALTGROUP command.
When you append the prefix **NO** to the name of the custom field, you delete the value for that custom field from the group's profile. For example, if your installation has defined a custom field named COMPADDR and you want to remove the COMPADDR field from the profile of the group ABCSUPPLY, you might issue the following command:

**Example:**

```
ALTGROUP ABCSUPPLY CSDATA(NOCOMPADDR)
```

**NOCSDATA**
Deletes the CSDATA segment from the group profile.

**DATA** | **NODATA**
---|---
**DATA('installation-defined-data')**
Specifies up to 255 characters of installation-defined data to be stored in the group profile and must be enclosed in single quotation marks. It can also contain double-byte character set (DBCS) data.

Use the LISTGRP command to list this information.

**NODATA**
Specifies that the ALTGROUP command is to delete any installation-defined data in the group profile.

**DFP** | **NODFP**
---|---
**DFP**
Specifies that when you change the profile of a group, you can enter any of the following operands to add, change, or delete default values for the DFP data application, data class, management class, and storage class. DFP uses this information to determine data management and DASD storage characteristics when a user creates a new data set for a group.

**DATAAPPL** | **NODATAAPPL**
---|---
**DATAAPPL(application-name)**
Specifies the name of a DFP data application. The name you specify can contain up to 8 alphanumeric characters.

**NODATAAPPL**
Specifies that you want to delete the DFP data application name from the DFP segment of the group's profile.

**DATACLAS** | **NODATACLAS**
---|---
**DATACLAS(data-class-name)**
Specifies the default data class. The class name you specify can contain up to 8 alphanumeric characters.

A data class can specify some or all of the physical data set attributes associated with a new data set. During new data set allocation, data management uses the value you specify as a default unless it is preempted by a higher priority default, or overridden in some other way (for example, by JCL).

**Note:** The value you specify must be a valid data class name defined for use on your system. For more information, see **z/OS Security Server RACF Security Administrator's Guide**.

For information on defining DFP data classes, see **z/OS DFSMSdfp Storage Administration**.
NODATACLAS
Specifies that you want to delete the default data class name from the DFP segment of the group’s profile.

MGMTCLAS | NOMGMTCLAS

MGMTCLAS(\textit{management-class-name})
Specifies the default management class. The class name you specify can contain up to 8 alphanumeric characters.

A management class contains a collection of management policies that apply to data sets. Data management uses the value you specify as a default unless it is preempted by a higher priority default, or overridden in some other way (for example, by JCL).

\textbf{Note:} The value you specify must be defined as a profile in the MGMTCLAS general resource class, and the group must be granted at least READ access to the profile. Otherwise, RACF does not allow the group access to the specified MGMTCLAS. For more information, see \textit{z/OS Security Server RACF Security Administrator’s Guide}.

For information on defining DFP management classes, see \textit{z/OS DFSMSdfp Storage Administration}.

NOMGMTCLAS
Specifies that you want to delete the default management class name from the DFP segment of the group’s profile.

STORCLAS | NOSTORCLAS

STORCLAS(\textit{storage-class-name})
Specifies the default storage class. The class name you specify can contain up to 8 alphanumeric characters.

A storage class specifies the service level (performance and availability) for data sets managed by the Storage Management Subsystem (SMS). During new data set allocation, data management uses the value you specify as a default unless it is preempted by a higher priority default, or overridden in some other way (for example, by JCL).

\textbf{Note:} The value you specify must be defined as a profile in the STORCLAS general resource class, and the group must be granted at least READ access to the profile. Otherwise, RACF does not allow the group access to the specified STORCLAS. For more information, see \textit{z/OS Security Server RACF Security Administrator’s Guide}.

For information on defining DFP storage classes, see \textit{z/OS DFSMSdfp Storage Administration}.

NOSTORCLAS
Specifies that you want to delete the default storage class name from the DFP segment of the group’s profile.

NODFP
Specifies that RACF should delete the DFP segment from the group’s profile.

MODEL | NOMODEL
ALTGROUP

MODEL(dsname)
   Specifies the name of a data set profile that RACF is to use as a model when new data set profiles are created that have group-name as the high-level qualifier. For this operand to be effective, the MODEL(GROUP) option on the SETROPTS command must be active. If the ALTGROUP command cannot find the dsname profile, it issues a warning message and places the profile name in the group entry.

   RACF always prefixes dsname with the group name when it accesses the profile.

   For information about automatic profile modeling, refer to z/OS Security Server RACF Security Administrator's Guide.

NOMODEL
   Specifies that the ALTGROUP command is to delete the model name in the group profile.

OMVS | NOOMVS

OMVS
   Specifies z/OS UNIX System Services information for the group profile being changed.

AUTOGID | GID | NOGID
   Specifies whether RACF is to automatically assign an unused GID value to the group, if a specific GID value is to be assigned or if the group identifier from the OMVS segment of the group’s profile is to be deleted.

   AUTOGID
   Specifies that RACF is to automatically assign an unused GID value to the group. The GID value is derived from information obtained from the BPX.NEXT.USER profile in the FACILITY class. For more information on setting up BPX.NEXT.USER, see z/OS Security Server RACF Security Administrator's Guide.

   If you are using RRSF automatic command direction for the GROUP class, the command sent to other nodes will contain an explicit assignment of the GID value which was derived by RACF on the local node.

   Rules:
   • AUTOGID cannot be specified if more than one group is entered.
   • The AUTOGID keyword is mutually exclusive with the SHARED keyword.
   • If both GID and AUTOGID are specified, AUTOGID is ignored.
   • If both NOGID and AUTOGID are specified, AUTOGID is ignored.
   • Field-level access checking for the GID field applies when using AUTOGID.
   • AUTOGID cannot be used to reassign a GID value if one already exists for the group. If AUTOGID is specified, but the group already has a GID assigned, one of two things will happen.
     – If the preexisting GID is unique to this group, this value will be identified in informational message IRR52177I, and the value will be left unchanged. If RRSF automatic command direction is in effect for the GROUP class, then the outbound ALTGROUP command will be altered to contain the preexisting GID value in the OMVS GID keyword.
ALTGROUP

- If the preexisting GID is not unique to this group, error message IRR52178I will be issued, and the command will fail. See IRR52178I for information on changing the group's existing GID value.

**GID(group-identifier) [SHARED]**

**GID(group-identifier)**

Specifies the group identifier. The GID is a numeric value from 0–2,147,483,647.

When a GID is assigned to a group, all users connected to that group who have a user identifier (UID) in their user profile can use functions such as the TSO/E command, OMVS, and can access z/OS UNIX files based on the GID and UID values assigned.

**Notes:**

1. If the security administrator has defined the SHARED.IDS profile in the UNIXPRIV class, the GID must be unique. Use the SHARED keyword in addition to GID to assign a value that is already in use.

2. If SHARED.IDS is not defined, RACF does not require the GID to be unique. The same value can be assigned to multiple groups, but this is not recommended because individual group control would be lost. However, if you want a set of groups to have exactly the same access to z/OS UNIX resources, you might decide to assign the same GID to more than one group.

3. RACF allows you to define and connect a user to more than 300 groups (which is the same as the NGROUPS_MAX variable defined in the POSIX standard), but when a process is created or z/OS UNIX group information is requested, only up to the first 300 z/OS UNIX groups are associated with the process or user.

   The first 300 z/OS UNIX groups that have GIDs to which a user is connected are used by z/OS UNIX. LISTUSER displays the groups in the order that RACF examines them when determining which of the user's groups are z/OS UNIX groups.

   See [z/OS UNIX System Services Planning](z/OS UNIX System Services Planning) for information on NGROUPS_MAX.

**SHARED**

If the security administrator has chosen to control the use of shared GIDs, this keyword must be used in addition to the GID keyword to specify the group identifier if it is already in use by at least one other group. The administrator controls shared GIDs by defining the SHARED.IDS profile in the UNIXPRIV class.

**Rules:**

- If the SHARED.IDS profile is not defined, SHARED is ignored.
- If SHARED is specified in the absence of GID, it is ignored.
ALTGROUP

• If the SHARED.IDS profile is defined and SHARED is specified, but the value specified with GID is not currently in use, SHARED is ignored and UNIXPRIV authority is not required.
• Field- level access checking for the GID field applies when using SHARED.
• The SHARED keyword is mutually exclusive with the AUTOGiD keyword.

NOGID
Specifies that you want to delete the group identifier from the OMVS segment of the group’s profile.

NOOMVS
Specifies that RACF delete the OMVS segment from the group’s profile.

OVM | NOOVM

OVM
Specifies OpenExtensions VM information for the group profile being changed.

GID | NOGID

GID(group-identifier)
Specifies the group identifier. The GID is a numeric value from 0–2 147 483 647.

Notes:
1. RACF does not require the GID to be unique. The same value can be assigned to multiple groups, but this is not recommended because individual group control would be lost. However, if you want a set of groups to have exactly the same access to the OpenExtensions VM resources, you might decide to assign the same GID to more than one group.
2. Exercise caution when changing the GID for a group. The following situations might occur:
   • If the file system contains files that contain the old GID as the file owner GID, the members of the group lose access to those files, depending on the permission bits associated with the file.
   • If files exist with an owner GID equal to the group’s new GID value, the members of the group gain access to these files.
   • If another group is subsequently added with the old value as its GID, the members of the group might have access to the old files.
   • If you have an EXEC.Ggid profile in the VMPOSIX class for the old GID value, make sure you delete this profile and create another to reflect the new value.
3. The value defined for the NGROUPS_MAX variable in the ICHNGMAX macro on VM defines the maximum number of OpenExtensions VM groups to be associated with an OpenExtensions VM process or user. The NGROUPS_MAX variable on VM is a number 32–125, inclusive. However, RACF allows you to define and connect a user to more than the number of groups defined in this variable. If the NGROUPS_MAX variable is \( n \) and a process is created or
OpenExtensions VM group information is requested, only up to the first \( n \) OpenExtensions VM groups are associated with the process or user. The first \( n \) OpenExtensions VM groups to which a user is connected are used by OpenExtensions VM.

LISTUSER displays the groups in the order that RACF examines them when determining which of the user's groups are OpenExtensions VM groups.

See [z/OS Security Server RACF Macros and Interfaces](#) for information on NGROUPS_MAX.

NOGID
Specifies that you want to delete the group identifier from the OVM segment of the group's profile.

If NOGID is specified for the group, the default GiD of 4294967295 (X'FFFFFFFF') is assigned on VM. The LISTGRP command displays the field name followed by the word NONE.

NOOVM
Specifies that RACF delete the OVM segment from the group’s profile.

OWNER(userid or group-name)
Specifies a RACF-defined user or group you want to be the new owner of the group.

To change the owner of a group, you must be the current owner of the group, or have the SPECIAL attribute, or have the group-SPECIAL attribute in the group owning the profile.

If you specify a group name, then OWNER and SUPGROUP must specify the same group name.

SUPGROUP(group-name)
Specifies the name of the RACF-defined group you want to make the new superior group for the group profile you are changing.

The new superior group must not be the same as the current one, and it must not have any level of subgroup relationship to the group you are changing.

To change a superior group, you must have the SPECIAL attribute, the group profile must be within the scope of a group in which you have the group-SPECIAL attribute, or you must have JOIN authority in, or be the owner of, both the current and new superior groups. Note that you can have JOIN authority in one group and be the owner of or have the group-SPECIAL attribute in the other group.

If owner is a group name, OWNER and SUPGROUP must specify the same group name.

TERMUACC | NOTERMUACC

TERMUACC
Specifies that during terminal authorization checking, RACF is to allow the use of the universal access authority for a terminal when it checks whether a user in the group is authorized to access a terminal.

NOTERMUACC
Specifies that the group or a user connected to the group must be authorized (using the PERMIT command with at least READ authority) to access a terminal.
### ALTGROUP

**TME**  
Specifies that information for the Tivoli Security Management Application is to be added, changed, or deleted.

**Note:** The TME segment fields are intended to be updated only by the Tivoli Security Management Application, which manages updates, permissions, and cross references. A security administrator should only directly update Tivoli Security Management fields on an exception basis.

**ROLES | ADDROLES | DELROLES | NOROLES**

**ROLES**(profile-name)  
Specifies a list of roles that reference this group.

The *profile-name* value should be the name of a defined role, which is a discrete general resource profile in the ROLE class.

**ADDROLES**(profile-name)  
Specifies a list of roles that reference this group.

The *profile-name* value should be the name of a defined role, which is a discrete general resource profile in the ROLE class.

**DELROLES**(profile-name)  
Specifies that specific roles from the current list of roles are to be removed.

The *profile-name* value should be the name of a defined role, which is a discrete general resource profile in the ROLE class.

**NOROLES**  
Specifies that the entire list of roles be removed.

**NOTME**  
Specifies that RACF delete the TME segment from the group profile.

### Examples

**Table 7. ALTGROUP examples**

**Example 1**

**Operation**  
User WJB10 wants to change the superior group and owning group for PROJECTA from RESEARCH to PAYROLL. Users connected to group PROJECTA are authorized access to terminals according to the universal access authority of the terminal.

**Known**  
User WJB10 has JOIN authority in RESEARCH and is the owner of PAYROLL.

PROJECTA is a subgroup of RESEARCH.

User WJB10 wants to issue the command as a RACF TSO command.

**Command**  
`ALTGROUP PROJECTA SUPGROUP(PAYROLL) OWNER(PAYROLL) TERMUACC`

**Defaults**  
None.

**Example 2**

**Operation**  
User MULES wants to change the superior group for PROJECTB from SYS1 to RESEARCH and assign RESEARCH as the new owner.

**Known**  
User MULES has the SPECIAL attribute.

PROJECTB is a subgroup of SYS1. User MULES wants to issue the command as a RACF operator command, and the RACF subsystem prefix is @.

**Command**  
`@ALTGROUP PROJECTB SUPGROUP(RESEARCH) OWNER(RESEARCH)`

**Defaults**  
None.
### Table 7. ALTGROUP examples (continued)

#### Example 3
**Operation** User SJR2 wants to change the installation-defined information associated with the RSC1 group and delete the model name. User SJR2 wants to direct the command to run under the authority of user ANW01.

**Known** User SJR2 is the owner of group RSC1. User SJR2 wants to issue the command as a RACF TSO command. SJR2 and ANW01 have an already established user ID association. User ANW01 is the owner of group RSC1.

**Command**
```
ALTGROUP RSC1 DATA('RESOURCE USAGE ADMINISTRATION') NOMODEL AT(.ANW01)
```

**Defaults** Command direction defaults to the local node.

#### Example 4
**Operation** User BILLC wants to make the following changes to the profile for group PROJECT6.
- Change the default DFP management class to MCLASS7
- Change the default DFP storage class to SCLASS3
- Change the default DFP data class to DCLASS15
- Delete the default DFP data application.

**Known**
- User BILLC has the SPECIAL attribute.
- Group PROJECT6 has been defined to RACF, and PROJECT6’s group profile contains a DFP segment.
- MCLASS7 has been defined to RACF as a profile in the MGMTCLAS general resource class, and group PROJECT6 has been given READ access to this profile.
- SCLASS3 has been defined to RACF as a profile in the STORCLAS general resource class, and group PROJECT6 has been given READ access to this profile.
- User BILLC wants to issue the command as a RACF TSO command.

**Command**
```
ALTGROUP PROJECT6 DFP(MGMTCLAS(MCLASS7) STORCLAS(SCLASS3) DATACLAS(DCLASS15) NODATAAPPL))
```

**Defaults** None.
ALTUSER

ALTUSER (Alter user profile)

Purpose

Use the ALTUSER command to change the information in a user's profile, including the user's system-wide attributes and authorities. The user profile consists of a RACF segment and, optionally, other segments such as a TSO segment or a DFP segment. You can use this command to change information in any segment of the user's profile.

When you change a user's level of authority in a group (using the AUTHORITY operand), RACF updates the appropriate group profile. When you change a user's default universal access authority for a group (using the UACC operand), RACF changes the appropriate connect profile. For all other changes, RACF changes the user's profile.

Note: If the user is currently logged on, changes to the attributes (except for OWNER and AUTHORITY) do not take effect until the next time the user logs on, even though the LISTUSER command shows the new values.

Attention:
- When the ALTUSER command is issued from ISPF, the TSO command buffer (including password and password phrase data) is written to the ISPLOG data set. As a result, you should not issue this command from ISPF or you must control the ISPLOG data set carefully.
- If the ALTUSER command is issued as a RACF operator command, the command and all data (including password and password phrase data) is written to the system log. Therefore, use of ALTUSER as a RACF operator command should either be controlled or you should issue the command as a TSO command.

Note that you cannot:
- Use the ALTUSER command to change a user ID association; you must use the RACLINK command.
- Use the ALTUSER command for profiles in the DIGTCERT class.
- Use the ALTUSER command for user IDs that have mixed-case characters, such as irrcerta, irrsitec, and irrmulti (which are associated with digital certificates).

Issuing options

The following table identifies the eligible options for issuing the ALTUSER command:

<table>
<thead>
<tr>
<th>As a RACF TSO command?</th>
<th>As a RACF operator command?</th>
<th>With command direction?</th>
<th>With automatic command direction?</th>
<th>From the RACF parameter library?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

For information on issuing this command as a RACF TSO command, refer to Chapter 3, “RACF TSO commands,” on page 15.

For information on issuing this command as a RACF operator command, refer to Chapter 4, “RACF operator commands,” on page 21.
ALTUSER

You must be logged on to the console to issue this command as a RACF operator command.

Related commands

- To add a user profile, see ADDUSER (Add user profile).
- To delete a user profile, see DELUSER (Delete user profile).
- To display information from a user profile, see LISTUSER (List user profile).
- To administer user ID associations, see RACLINK (Administer user ID associations).
- To obtain a list of user profiles, see SEARCH (Search RACF database).

Authorization required

When issuing the ALTUSER command as a RACF operator command, you might require sufficient authority to the proper resource in the OPERCMDS class. See z/OS Security Server RACF Security Administrator’s Guide for further information.

The level of authority required depends on which of the user’s attributes you want to change.

- If you have the SPECIAL attribute, you can use all the operands except UAUDIT/NOUAUDIT.
- To specify the AT keyword, you must have READ authority to the DIRECT.node resource in the RRSFDATA class and a user ID association must be established between the specified node.userid pair(s).
- To specify the ONLYAT keyword you must have the SPECIAL attribute, the userid specified on the ONLYAT keyword must have the SPECIAL attribute, and a user ID association must be established between the specified node.userid pair(s) if the user IDs are not identical.
- If the owner of the user profile is within the scope of a group in which you have the group-SPECIAL attribute, you can use all of the operands except SPECIAL, AUDITOR, OPERATIONS, NOEXPIRED and UAUDIT/NOUAUDIT.
- If you are the owner of the user’s profile, you can use any of the following operands for user-related attributes:
  - ADSP | NOADSP
  - DATA | NODATA
  - DFLTGRP
  - GRPACC | NOGRPACC
  - MODEL | NOMODEL
  - NAME
  - OIDCARD | NOOIDCARD
  - OWNER
  - PASSWORD | NOPASSWORD
  - PHRASE | NOPHRASE
  - RESTRICTED | NORESTRICTED
  - RESUME | NORESUME
  - REVOKE | NOREVOKE
  - WHEN
- Each user can change his or her name field or default group (NAME and DFLTGRP operands). If an installation specifies MODEL(USER) on the SETROPTS command, each user can also change his or her model data set profile name (using the MODEL operand).
- You can use the ADSP | NOADSP, RESTRICTED | NORESTRICTED, GROUP, AUTHORITY, and UACC operands for group-related user attributes if you have
JOIN or CONNECT authority, or if the group profile is within the scope of a group in which you have the group-SPECIAL attribute, or if you are the owner of the specified group.

- To specify the AUDITOR/NOAUDITOR, SPECIAL/NOSPECIAL, and OPERATIONS/NOOPERATIONS operands as system-wide user attributes, you must have the SPECIAL attribute.
- To specify the UAUDIT/NOUAUDIT operand, either you must have the AUDITOR attribute, or the user profile must be within the scope of a group in which you have the group-AUDITOR attribute.
- You can specify the CLAUTH and NOCLAUTH operands if you are the owner of the user's profile and have the CLAUTH attribute for the class to be added or deleted.
- To assign a security category to a profile, or to delete a category from a profile, you must have the SPECIAL attribute, or the category must be in your user profile.
- To assign a security level to a profile, or to delete a security level from a profile, you must have the SPECIAL attribute, or, in your own profile, a security level that is equal to or greater than the security level you are assigning or deleting.
- To assign or delete a security label when SETROPTS SECLABELCONTROL is active, you must have the SPECIAL attribute.
- To change information within a segment other than the base segment, you must have one of the following:
  - The SPECIAL attribute
  - At least UPDATE authority to the desired field within the segment through field-level access control.


- To reset passwords and password phrases or to resume user IDs, you must have at least one of the following authorizations:
  - You have the SPECIAL attribute.
  - You have group-SPECIAL authority over the user profile.
  - You are the OWNER of the user profile.
  - You have sufficient access to the IRR.PASSWORD.RESET resource in the FACILITY class.
  - You have sufficient access to an appropriate resource in the FACILITY class (IRR.PWRESET.OWNER.OWNER.owner or IRR.PWRESET.TREE.owner), and both of the following conditions are also true:
    - The other user does not have the SPECIAL, OPERATIONS, AUDITOR, or PROTECTED attribute.
    - You are not excluded from altering the user by the IRR.PWRESET.EXCLUDE.EXCLUDED-USER resource in the FACILITY class.


When your reset and resume authority is through your access to the IRR.PASSWORD.RESET resource, the IRR.PWRESET.OWNER.OWNER.owner resource, or the IRR.PWRESET.TREE.owner resource, the following requirements apply:

- If you have READ access, you can:
  - Use the PASSWORD operand to reset a password (to an expired password) for a user who does not have the SPECIAL, OPERATIONS, AUDITOR, or PROTECTED attribute.
ALTUSER

- Use the PHRASE operand to reset a password phrase (to an expired password phrase) for a user with an assigned password phrase who does not have the SPECIAL, OPERATIONS, AUDITOR, or PROTECTED attribute. **Note**: You cannot use the PHRASE operand to *add* a password phrase for a user who does not have one.
- Use the RESUME operand, without specifying a date, for a user who does not have the SPECIAL, OPERATIONS, AUDITOR, or PROTECTED attribute.
  - If you have UPDATE access, you can:
    - Use the PASSWORD, PHRASE, and RESUME operands as noted for READ access.
    - Use the NOEXPIRED operand (with PASSWORD or PHRASE) for a user who does not have the SPECIAL, OPERATIONS, AUDITOR, or PROTECTED attribute.
  - If you have CONTROL access, you can:
    - Use the PASSWORD, PHRASE, RESUME, and NOEXPIRED operands as noted for READ and UPDATE access.
    - Reset the password or password phrase within the minimum change interval for a user who does not have the SPECIAL, OPERATIONS, AUDITOR, or PROTECTED attribute.
- To specify the SHARED keyword, you must have the SPECIAL attribute or at least READ authority to the SHARED.IDS resource in the UNIXPRIV class.

Syntax

For the key to the symbols used in the command syntax diagrams, see "Syntax of RACF commands and operands" on page 9. The complete syntax of the ALTUSER command is:

```
[subsystem-prefix][ALTUSER | ALU]
  (userid ...)
    [ ADDCATEGORY(category-name ...)]
    [ DELCATEGORY [(category-name ... | *)]
    [ ADSP | NOADSP ]
    [ AT([node],userid ...) | ONLYAT([node],userid ...) ]
    [ AUDITOR | NOAUDITOR ]
    [ AUTHORITY(group-authority) ]
    [ CICS( ]
      [ OPCLASS(operator-class ...)]
      [ ADDOPCLASS(operator-class ...)]
      [ DELOPCCLASS(operator-class ...)]
      [ NOOPCLASS ]
      [ OPIDENT(operator-id) | NOOPIENT ]
      [ OPERY(operator-priority) | NOOPRTY ]
      [ RSLKEY(rslkey ... | 0 | 99) | NORSLLKEY ]
      [ TIMEOUT(timeout-value) | NOTIMEOUT ]
      [ TSLKEY(tslkey ... | 0 | 1 | 99) | NOTSLKEY ]
      [ XRFSOFF(Force | NOFORCE) | NOXRFSOFF ]
    )
    [ NOCICS ]
    [ [CIAUTH | NOCLAUTH] (class-name ...) ]
    [ CSDATA( ]
      [ custom-field-name(custom-field-value) | NOcustom-field-name ] ...
    )
    [ NOCSDATA ]
```
ALTUSER

[ DATA('installation-defined-data') ]
[ DCE(
    [ AUTOLOGIN(YES | NO) ]
    [ DCENAME(user-principal-name) ]
    [ HOMECELL(dce-cell-name) ]
    [ HOMEUUID(home-cell-UUID) ]
    [ UUID(universal-unique-identifier) ]
  )
  ]
[ DFLTGRP(group-name) ]
[ DFP(
    [ DATAAPPL(application-name) ]
    [ DATACLAS(data-class-name) ]
    [ MGMTCLAS(management-class-name) ]
    [ STORCLAS(storage-class-name) ]
  )
  ]
[ EIM(
    [ LDAPPROF(ldapbind_profile) ]
  )
[ EXPIRED | NOEXPIRED ]
[ GROUP(group-name) ]
[ GRPACC | NOGRPACC ]
[ KERB(
    [ ENCRYPT (
      [ DES | NODES ]
      [ DES3 | NODES3 ]
      [ DESD | NODESD ]
      [ AES128 | NOAES128 ]
      [ AES256 | NOAES256 ]
    )
  )]
[ LANGUAGE(
    [ PRIMARY(language) ]
    [ SECONDARY(language) ]
  )]
[ LNOTES(
    [ SNAME(short-name) ]
  )]
[ MODEL(dsname) ]
[ NAME(user-name) ]
[ NDS(
    [ UNAME(user-name) ]
  )]
[ NONDS ]
Chapter 5. RACF command syntax
For information on issuing this command as a RACF TSO command, refer to Chapter 3, “RACF TSO commands,” on page 15.
ALTUSER

Parameters

subsystem-prefix
Specifies that the RACF subsystem is the processing environment of the command. The subsystem prefix can be either the installation-defined prefix for RACF (1–8 characters) or, if no prefix has been defined, the RACF subsystem name followed by a blank. If the command prefix was registered with CPF, you can use the MVS command D OPDATA to display it or you can contact your RACF security administrator.

Only specify the subsystem prefix when issuing this command as a RACF operator command. The subsystem prefix is required when issuing RACF operator commands.

userid
Specifies the RACF-defined user or users whose profile you want to change. If you specify more than one user ID, the list must be enclosed in parentheses. This operand is required and must be the first operand following ALTUSER.

ADDCATEGORY | DELCATEGORY

ADDCATEGORY(category-name)
Specifies one or more names of installation-defined security categories. The names you specify must be defined as members of the CATEGORY profile in the SECDATA class. For information on defining security categories, see z/OS Security Server RACF Security Administrator’s Guide.

When the SECDATA class is active and you specify ADDCATEGORY, RACF performs security category checking in addition to its other authorization checking. If a user requests access to a data set, RACF compares the list of security categories in the user profile with the list of security categories in the data set profile. If RACF finds any security category in the data set profile that is not in the user’s profile, RACF denies access to the data set. If the user’s profile contains all the required security categories, RACF continues with other authorization checking.

Note: RACF does not perform security category checking for a started task or user that has the RACF privileged or trusted attribute. The RACF privileged or trusted attribute can be assigned to a started task through the RACF started procedures table or STARTED class, or to other users by installation-supplied RACF exits.

DELCATEGORY[(category-name...|*)]
Specifies one or more names of the installation-defined security categories you want to delete from the user profile. Specifying an asterisk (*) deletes all categories; the user no longer has access to any resources protected by security category checking.

Specifying DELCATEGORY without category-name causes RACF to delete only undefined category names (those names that once were valid names but that the installation has since deleted from the CATEGORY profile).

ADSP | NOADSP

ADSP
Assigns the ADSP attribute to the user. This means that all permanent tape and DASD data sets the user creates are automatically RACF-protected by
ALTUSER

discrete profiles. ADSP specified on the ALTUSER command overrides NOADSP specified on the CONNECT command.

The ADSP attribute has no effect (even if assigned to a user) if SETROPTS NOADSP is in effect.

NOADSP

Specifies that the user no longer has the ADSP attribute.

AT | ONLYAT

The AT and ONLYAT keywords are only valid when the command is issued as a RACF TSO command.

**AT**([node].userid ...)

Specifies that the command is to be directed to the node specified by node, where it runs under the authority of the user specified by userid in the RACF subsystem address space.

If node is not specified, the command is directed to the local node.

**ONLYAT**([node].userid ...)

Specifies that the command is to be directed only to the node specified by node where it runs under the authority of the user specified by userid in the RACF subsystem address space.

If node is not specified, the command is directed only to the local node.

AUDITOR | NOAUDITOR

**AUDITOR**

Specifies that the user is to have full responsibility for auditing the use of system resources. An AUDITOR user can control the logging of detected accesses to any RACF-protected resources during RACF authorization checking and accesses to the RACF database.

You must have the SPECIAL attribute to enter the AUDITOR operand.

**NOAUDITOR**

Specifies that the user no longer has the AUDITOR attribute.

You must have the SPECIAL attribute to enter the NOAUDITOR operand.

**AUTHORITY**(group-authority)

Specifies the new level of authority the user is to have in the group specified in the GROUP operand. The valid group authority values are USE, CREATE, CONNECT, and JOIN, as described in "Group authorities" on page 13. If you specify AUTHORITY without group-authority, RACF ignores the operand and the existing group authority remains unchanged.

CICS | NOCICS

Adds, alters, or deletes CICS operator information for a CICS terminal user.

If you are adding a CICS segment to a user profile, omitting a suboperand is equivalent to omitting the suboperand on the ADDUSER command. If you are changing an existing CICS segment in a user profile, omitting a suboperand leaves the existing value for that suboperand unchanged.

You can control access to the entire CICS segment or to individual fields within the CICS segment by using field-level access checking. For more information, see [z/OS Security Server RACF Security Administrator's Guide](https://www.ibm.com/support/knowledgecenter/en/SSS57D_11.2.0/com.ibm.zos.racf.secmgmt.rexecd/).

**OPCLASS | ADDOPCLASS | DELOPCLASS | NOOPCLASS**

Where operator-class1, operator-class2 are numbers in the range 1–24,
defined as two digits. These numbers represent classes assigned to this
operator to which BMS (basic mapping support) messages are routed.

**OPCLASS**(operator-class ...)
Specifies the list of classes assigned to this operator to which BMS
messages are routed.

**ADDOCLASS**(operator-class ...)
Adds to the list of classes assigned to this operator to which BMS
messages are routed.

**DELOCLASS**(operator-class ...)
Deletes only the specified classes from the list of classes assigned to
this operator to which BMS messages are routed.

**NOOPCLASS**
Deletes all operator classes from this profile and returns the user to the
CICS defaults for this field. This field no longer appears in LISTUSER
output.

**OPIDENT | NOOPIDENT**

**OPIDENT**(operator-id)
Specifies a 1–3 character identification of the operator for use by BMS.

Operator identifiers can consist of any characters, and can be entered
with or without single quotation marks. The following rules apply:

- If parentheses, commas, blanks, or semicolons are to be entered as
  part of the operator identifier, the character string must be enclosed
  in single quotation marks. For example, if the operator identifier is
  (1), you must enter **OPIDENT('1')**.
- If a single quotation mark is intended to be part of the operator
  identifier, use two single quotation marks together for each single
  quotation mark within the string, and enclose the entire string within
  single quotation marks.

**NOOPIDENT**
Deletes the operator identification and returns the user to the CICS
default for this field. The **OPIDENT** field defaults to blanks in the RACF
user profile, and blanks appear for the field in LISTUSER output.

**OPPRTY | NOOPPRTY**

**OPPRTY**(operator-priority)
Specifies a number in the range 0–255 that represents the priority of
the operator.

**NOOPPRTY**
Deletes the operator priority and returns the user to the CICS default for
this field.

This field defaults to zeros in the RACF user profile, and zeros appear
for the field in LISTUSER output.

**RSLKEY | NORSKEY**

**RSLKEY**(rskey ... | 0 | 99)
Specifies the complete list of resource security level (RSL) keys
assigned to the user. The RSL keys are used by CICS on distributed
platforms. Each CICS resource has one RSL key assigned to it; in
order for a user to access a resource, the user must have the same
RSL key as the RSL key assigned to the resource.
RSLKEY does not add or delete keys. It only replaces existing keys. Use NORSLKEY to delete keys.

- RSLKEY(rslkey ...) specifies a list of one or more numbers in the range of 1–24 which represent the resource security level (RSL) keys assigned to the user.
- If RSLKEY(0) is specified, no RSL keys are assigned to the user.
- If RSLKEY(99) is specified, all RSL keys are assigned to the user (1–24, inclusive).
- Keys 0 and 99 are mutually exclusive and cannot be specified with any other keys.
- If RSLKEY is specified with no key numbers, RSLKEY(0) is defaulted.

NORSLKEY
Specifies that you want to remove the RSL key list from the user's RACF user profile. CICS will treat it as RSLKEY(0).

TIMEOUT | NOTIMEOUT

TIMEOUT(timeout-value)
Specifies the time, in hours and minutes, that the operator is allowed to be idle before being signed off. The value for TIMEOUT can be entered in the form m, mm, hmm, or hhmm, where the value for m or mm is 00–59, or 00–60 if h or hh is not specified or is specified as 0 or 00. The value for h or hh must be 00–99.

If this suboperand is omitted, there is no change to this field.

NOTIMEOUT
Deletes the timeout value and returns the user to the CICS default for this field.

This field defaults to zeros in the RACF user profile, and zeros appear for the field in LISTUSER output.

TSLKEY | NOTSLKEY

TSLKEY(tslkey ... | 0 | 1 | 99)
Specifies the complete list of transaction security level (TSL) keys assigned to the user. The TSL keys are used by CICS on distributed platforms. Each CICS transaction has one TSL key assigned to it; in order for a user to run a transaction, the user must have the same TSL key as the TSL key assigned to the transaction.

TSLKEY does not add or delete keys. It only replaces existing keys. Use NOTSLKEY to delete keys.

- TSLKEY(tslkey ...) specifies a list of one or more values of 1–64 which represent the transaction security level (TSL) keys assigned to the user.
- If TSLKEY(0) is specified, no TSL keys are assigned to the user.
- If TSLKEY(99) is specified, all TSL keys are assigned to the user (1–64, inclusive).
- Keys 0 and 99 are mutually exclusive and cannot be specified with any other keys.
- If TSLKEY is specified with no key numbers, TSLKEY(1) is defaulted.
NOTSLKEY
  Specifies that you want to remove the TSL key list from the user's
  RACF user profile. CICS will treat it as TSLKEY(1).

XRFSOFF | NOXRFSOFF
  XRFSOFF(FORCE | NOFORCE)
    Specifies that the user is to be signed off by CICS when an XRF
    takeover occurs.

  NOXRFSOFF
    Returns the user to the CICS default for this field.
    This field defaults to NOFORCE in the RACF user profile, and
    NOFORCE appears in LISTUSER output.

NOCICS
  Deletes the CICS segment from a user profile. No CICS information
  appears in LISTUSER output.

CLAUTH | NOCLAUTH
  CLAUTH(class-name ...)
    Specifies the classes in which the user is allowed to define profiles to
    RACF for protection, in addition to the classes previously allowed for the
    user. Classes you can specify are USER, and any resource class defined in
    the class descriptor table. RACF adds the class names you specify to the
    class names previously specified for this user.
    
    To enter the CLAUTH operand, you must have the SPECIAL attribute, or
    the user's profile must be within the scope of a group in which you have the
    group-SPECIAL attribute and have the CLAUTH attribute, or you must be
    the owner of the user's profile and have the CLAUTH attribute for the class
    to be added.

    Note: The CLAUTH attribute has no meaning for the FILE and
          DIRECTORY classes.

  NOCLAUTH(class-name ...)
    Specifies that the user is not allowed to define profiles to RACF for the
    classes that you specify. Classes you can specify are USER and any
    resource class name defined in the user profile. RACF deletes the class
    names you specify from the class names previously allowed for this user.
    
    To enter the NOCLAUTH operand specifying a class in the class descriptor
    table, you must have the SPECIAL attribute, or the user's profile must be
    within the scope of a group in which you have the group-SPECIAL attribute
    and have the CLAUTH attribute, or you must be the owner of the user's
    profile and have the CLAUTH attribute for the class to be deleted.
    
    To enter the NOCLAUTH operand specifying a class that is not in the class
    descriptor table you must have the SPECIAL attribute.
    
    If you do not have sufficient authority for a specified class, RACF ignores the
    CLAUTH or NOCLAUTH specification for the class and continues processing
    with the next class name specified.

CSDATA | NOCSDATA
  CSDATA
    Specifies information to add, change, or remove a custom field for this user.

    custom-field-name ... | NOcustom-field-name ...
ALTUSER

`custom-field-name(custom-field-value) ...`

Specifies the name and value of a custom field for this user. You can specify values for multiple custom fields with a single ALTUSER command.

Usage for each custom field is defined using the CFDEF operand of the RDEFINE command for resource profiles in the CFIELD class. Contact your security administrator to see how custom fields are used at your installation. For more information about custom fields, see [z/OS Security Server RACF Security Administrator's Guide](#).

Rules:
- You must use the same `custom-field-name` as defined by the CFIELD profile named USER.CSDATA.`custom-field-name`. (The CFIELD profile is defined using the CFDEF operand of the RDEFINE command.)
- You must specify a `custom-field-value` that is valid for the attributes of this custom field. (The attributes, such as data type, are defined in the CFDEF segment of the CFIELD profile.)

`NOcustom-field-name ...`

Removes the custom field information for this user. You can remove values for multiple custom fields with a single ALTUSER command.

When you append the prefix `NO` to the name of the custom field, you delete the value for that custom field from the user's profile. For example, if your installation has defined a custom field named ADDRESS and you want to remove the ADDRESS field from the profile of the user SHANNON, you might issue the following command:

**Example:**

```
ALTUSER SHANNON CSDATA(NOADDRESS)
```

`NOCSDATA`

Deletes the CSDATA segment from the user profile.

`DATA | NODATA`

**DATA('installation-defined-data')**

Specifies up to 255 characters of installation-defined data to be stored in the user's profile and must be enclosed in single quotation marks. It can also contain double-byte character set (DBCS) data. Note that only 254 characters of data are available for installation exits. If your installation has exits that examine this data, you should specify a maximum of 254 characters.

Use the LISTUSER command to list this information.

**NODATA**

Specifies that the ALTUSER command is to delete the installation-defined data in the user's profile.

`DCE | NODCE`

**DCE**

Specifies that, when you define an z/OS DCE user to RACF, you can enter any of the following suboperands to specify information for that user. Each suboperand defines information that RACF stores in a field within the DCE segment of the user's profile.
You can control access to an entire DCE segment or to individual fields within the DCE segment by using field level access checking.

**AUTOLOGIN(YES | NO) | NOAUTOLOGIN**

Specifies whether z/OS UNIX DCE is to log this user into z/OS UNIX DCE automatically. If AUTOLOGIN(NO) or NOAUTOLOGIN is specified, z/OS UNIX DCE does not attempt to login this user to z/OS UNIX DCE automatically. If AUTOLOGIN is not specified, AUTOLOGIN(NO) is the default.

**DCENAME | NODCENAME**

**DCENAME(user-principal-name)**

Specifies the DCE principal name defined for this RACF user in the DCE registry.

The DCENAME you define to RACF can contain 1–1023 characters and can consist of any character. You can enter the name with or without single quotation marks, depending on the following:

- If parentheses, commas, blanks, or semicolons are entered as part of the name, the character string must be enclosed in single quotation marks.
- If a single quotation mark is intended to be part of the character string, use two single quotation marks together for each single quotation mark within the string, and enclose the entire string within single quotation marks.

Both uppercase and lowercase characters are accepted and maintained in the case in which they are entered. RACF does not ensure that a valid DCENAME has been specified.

The DCENAME assigned to a user must be the same as the DCE principal name defined to the DCE registry.

If DCENAME is not specified, the LISTUSER command does not display a DCENAME for this user.

**Note:** RACF does not enforce the uniqueness of each DCENAME. The DCENAME specified must match the user's DCE principal name that is defined to the DCE registry. If the DCENAME entered does not correspond to the DCE principal name entered in the DCE registry for this user, z/OS UNIX DCE cannot correctly associate the identity of the DCE principal with the correct RACF user ID.

**NODCENAME**

Specifies that you want to delete the DCE principal name from the DCE segment of the user's profile.

If NODCENAME is specified, the LISTUSER command does not display a DCENAME for this user.

**HOMECELL | NOHOMECELL**

**HOMECELL(dce-cell-name)**

Specifies the DCE cell name defined for this RACF user.

The HOMECELL you define to RACF can contain 1–1023 characters and can consist of any character. You can enter the name with or without single quotation marks, depending on the following:
If parentheses, commas, blanks, or semicolons are entered as part of the cell name, the character string must be enclosed in single quotation marks.

If a single quotation mark is intended to be part of the cell name, use two single quotation marks together for each single quotation mark within the string, and enclose the entire string within single quotation marks.

Both uppercase and lowercase characters are accepted and maintained in the case in which they are entered. The fully qualified pathname should be specified. RACF does not ensure that a valid DCE cell name has been specified.

The HOMECCELL assigned to a user must be the same as the DCE cell name that this user has been defined to.

If the HOMECCELL is not specified, z/OS UNIX DCE single signon to DCE support assumes that the HOMECCELL for this user is the same cell where this MVS system is defined.

RACF checks that the prefix of the HOMECCELL name entered has a prefix of either /.../ or /.:/. The notation /.../ indicates that the HOMECCELL name is a global domain name service (DNS) cell name or X.500 global name.

The notation /.:/ indicates that the HOMECCELL name is a cell relative CDS (cell directory service) name. When determining the naming conventions used within your DCE cell, you should contact your DCE cell administrator.

NOHOMECCELL

Specifies that you want to delete the cell information from the DCE segment of the user profile.

If NOHOMECCELL is specified, the LISTUSER command does not display the HOMECCELL for this user.

HOMEUUID | NOHOMEUUID

HOMEUUID(home-cell-UUID)

Specifies the DCE universal unique identifier (UUID) for the cell that this user is defined to. The UUID is a 36-character string that consists of numeric and hexadecimal characters. This string must have the delimiter character (-) in positions 9, 14, 19, and 24. The general format for the UUID string is xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx, in which x represents a valid numeric or hexadecimal character.

Be careful when assigning UUIDs. The UUID cannot be randomly assigned. The HOMEUUID is the DCE UUID of the cell that this RACF user is defined to. If HOMEUUID is not specified, the LISTUSER command displays NONE for the HOMEUUID field.

Note: The HOMEUUID specified must match the UUID of the DCE cell to which this principal (specified by the DCENAME operand) is defined.

NOHOMEUUID

Specifies that you want to delete the home cell unique universal identifier from the DCE segment of the user’s profile.
If NOHOMEUUID is specified, LISTUSER for that user ID shows NONE for the HOMEUUID field.

**UUID | NOUUID**

**UUID**(universal-unique-identifier)
Specifies the DCE universal unique identifier (UUID) of the DCE principal defined in DCENAME. The UUID is a 36-character string that consists of numeric and hexadecimal characters. This string must have the delimiter character (-) in positions 9, 14, 19, and 24. The general format for the UUID string is xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx; in which x represents a valid numeric or hexadecimal character.

Be careful when assigning UUIDs. The UUID cannot be randomly assigned.

The DCE UUID assigned to a user must be the same as the DCE UUID assigned when defining this RACF user to the DCE registry as a DCE principal.

If UUID is not specified, the user cannot become an z/OS DCE user and a LISTUSER command for that user ID shows NONE for the UUID.

**Note:** RACF does not enforce the uniqueness of each UUID entered. The UUID specified must match the UUID in the DCE registry for the principal (specified by the DCENAME operand) that is being cross-linked with this RACF user ID.

**NOUUID**
Specifies that you want to delete the DCE unique universal identifier from the DCE segment of the user’s profile.

If NOUUID is specified, LISTUSER for that user ID shows NONE for the UUID field.

**NODCE**
Specifies that RACF should delete the DCE segment from the user’s profile.

**DFLTGRP**(group-name)
Specifies the name of a RACF-defined group to be used as the new default group for the user. The user must already be connected to this new group with at least USE authority. The user remains connected to the previous default group.

**DFP | NODFP**

**DFP**
Specifies that when you change the profile of a user, you can enter any of the following suboperands to add, change, or delete default values for the DFP data application, data class, management class, and storage class. DFP uses this information to determine data management and DASD storage characteristics when a user creates a new data set.

You can control access to the entire DFP segment or to individual fields within the DFP segment by using field-level access checking. For more information, see [z/OS Security Server RACF Security Administrator's Guide](#).

**DATAAPPL | NODATAAPPL**
DATAAPPL(application-name)
Specifies the name of a DFP data application. The name you specify can contain up to 8 alphanumeric characters.

NODATAAPPL
Specifies that you want to delete the DFP data application name from the DFP segment of the user’s profile.

DATACLAS | NODATACLAS
DATACLAS(data-class-name)
Specifies the default data class. The class name you specify can contain up to 8 alphanumeric characters.

A data class can specify some or all of the physical data set attributes associated with a new data set. During new data set allocation, data management uses the value you specify as a default unless it is preempted by a higher priority default, or overridden in some other way (for example, by JCL).

The value you specify must be a valid data class name defined for use on your system. For more information, see z/OS Security Server RACF Security Administrator’s Guide.

For information on defining DFP data classes, see z/OS DFSMSdfp Storage Administration.

NODATACLAS
Specifies that you want to delete the default data class name from the DFP segment of the user’s profile.

MGMTCLAS | NOMGMTCLAS
MGMTCLAS(management-class-name)
Specifies the default management class. The class name you specify can contain up to 8 alphanumeric characters.

A management class contains a collection of management policies that apply to data sets. Data management uses the value you specify as a default unless it is preempted by a higher priority default, or overridden in some other way (for example, by JCL).

The value you specify must be defined as a profile in the MGMTCLAS general resource class, and the user must be granted at least READ access to the profile. Otherwise, RACF does not allow the user access to the specified MGMTCLAS. For more information, see z/OS Security Server RACF Security Administrator’s Guide.

For information on defining DFP management classes, see z/OS DFSMSdfp Storage Administration.

NOMGMTCLAS
Specifies that you want to delete the default management class name from the DFP segment of the user’s profile.

STORCLAS | NOSTORCLAS
STORCLAS(storage-class-name)
Specifies the default storage class. The class name you specify can contain up to 8 alphanumeric characters.

A storage class specifies the service level (performance and availability) for data sets managed by the storage management
subsystem (SMS). During new data set allocation, data management uses the value you specify as a default unless it is preempted by a higher priority default, or overridden in some other way (for example, by JCL).

The value you specify must be defined as a profile in the STORCLAS general resource class, and the user must be granted at least READ access to the profile. Otherwise, RACF does not allow the user access to the specified STORCLAS. For more information, see z/OS Security Server RACF Security Administrator’s Guide.

For information on defining DFP storage classes, see z/OS DFSMSdfp Storage Administration.

NOSTORCLAS
Specifies that you want to delete the default storage class name from the DFP segment of the user's profile.

NODFP
Specifies that RACF should delete the DFP segment from the user’s profile.

EIM | NOEIM
Specifies or deletes the bind information required to establish a connection with the EIM domain.

EIM
Specifies the EIM segment for the user’s profile.

LDAPROF(ldapbind_profile)
Specifies the name of a profile in the LDAPBIND class. The profile in the LDAPBIND class contains the name of an EIM domain and the bind information required to establish a connection with the EIM domain. The EIM services attempt to retrieve this information when it is not explicitly supplied through invocation parameters. Applications or other services that use the EIM services may instruct their callers to define a profile in the LDAPBIND class or the IRR.PROXY.DEFAULTS profile in the FACILITY class.

The ldapbind_profile specifies the name of a profile in the LDAPBIND class containing the EIM domain and the LDAP bind information. The ldapbind_profile name may be 1–246 characters long. It is not a case-sensitive name.

NOLDAPROF
Deletes the LDAPBIND profile name from the EIM segment in the user's profile.

NOEIM
Deletes the EIM segment from the user’s profile

EXPIRED | NOEXPIRED

EXPIRED
Specifies that the new password or password phrase (specified with the PASSWORD or PHRASE keyword) or the new password defaulted by the PASSWORD keyword is marked as expired. Specifying the EXPIRED keyword requires the user to change their new password or password phrase at the next logon or job start.

The EXPIRED keyword is only valid when specified with the PASSWORD or PHRASE keyword.
ALTUSER

When EXPIRED is specified with the PHRASE keyword, the password phrase you specify is subject to the basic RACF rules for password phrase syntax and to any rules set by the installation through the new-password-phrase exit (ICHPWX11), if present.

When EXPIRED is specified with the PASSWORD keyword, the password you specify is not subject to the password syntax rules set by the installation through the SETROPTS PASSWORD command, nor is the password checked by the new-password exit (ICHPWX01), if present.

NOEXPIRED

Specifies that the password specified by the PASSWORD keyword or the password phrase specified by the PHRASE keyword need not be changed at the next logon. The NOEXPIRED keyword is only valid when specified with the PASSWORD or PHRASE keyword. NOEXPIRED does not indicate that the password or password phrase never expires. If you wish to set a password or password phrase that never expires, use the NOINTERVAL keyword on the PASSWORD command.

When NOEXPIRED is specified, the password or password phrase value you supply is subject to certain rules. Those rules include the basic RACF rules for password phrase syntax and any rules set by the installation through the SETROPTS PASSWORD command. In addition, the new-password exit (ICHPWX01), if present, is called to check passwords. The new-password-phrase exit (ICHPWX11), if present, is called to check password phrases and perform additional validation.

To specify NOEXPIRED, you must either have the SPECIAL attribute (at the system level), or have UPDATE access to the IRR.PASSWORD.RESET resource in the FACILITY class. Being the owner of the USER profile or having the group-SPECIAL attribute is not sufficient when NOEXPIRED is specified.

GROUP(group-name)

Specifies the group to which changes to the group-related user attributes UACC and AUTHORITY are to be made. The user must be connected to the specified group.

If you omit GROUP, the changes apply to the user’s default group. If you omit GROUP and specify DFLTGRP, however, the changes still apply to the user’s previous default group.

GRPACC | NOGRPACC

GRPACC

Specifies that any group data sets protected by DATASET profiles defined by this user are automatically accessible to other users in the group. The group whose name is used as the high-level qualifier of the data set name (or the qualifier supplied by a command installation exit) has UPDATE access authority in the new profile. GRPACC specified on the ALTUSER command overrides NOGRPACC specified on the CONNECT command.

NOGRPACC

Specifies that the user no longer has the GRPACC attribute.

KERB | NOKERB

KERB

Specifies z/OS Integrated Security Services Network Authentication Service
information for a user defined to RACF. Each subkeyword defines information that RACF stores in a field within the KERB segment of the user's profile.

**Note:** The RACF user password must be changed to be non-expired in order to complete the definition of the z/OS Network Authentication Service principal. The user cannot use any z/OS Network Authentication Service function until the definition is complete.

**encrypt | noencrypt**

**encrypt**

Specifies which keys the user (the z/OS Network Authentication Service principal) is allowed to use.

**DES | NODES**

Whether DES encrypted keys can be used.

**DES3 | NODES3**

Whether DES3 encrypted keys can be used.

**DESd | NODESd**

Whether DESD encrypted keys can be used.

**AES128 | NOAES128**

Whether AES128 encrypted keys can be used.

**AES256 | NOAES256**

Whether AES256 encrypted keys can be used.

When a principal’s password changes, a key of each type is generated and stored in the principal’s user profile. The use of each key is based on the z/OS Network Authentication Service configuration.

**Important:** When you enable the use of a new key type, be sure that the principal's password is changed to ensure that a key of the new type is generated and stored in the principal's user profile.

See [z/OS Integrated Security Services Network Authentication Service Administration](#) for information about how z/OS Network Authentication Service uses keys and how to customize environment variables related to keys.

**noencrypt**

Specifies that there is no restriction on which generated keys the principal can use, and resets the KERB ENCRYPT values to the default settings.

See [z/OS Integrated Security Services Network Authentication Service Administration](#) for information about how z/OS Network Authentication Service uses keys and how to customize environment variables related to keys.

**kerbname | nokerbname**

**kerbname(ke rberos-principal-name)**

Specifies the z/OS user ID's local kerberos-principal-name.

The value specified for the local kerberos-principal-name must be unique. Consequently, a list of users cannot be specified on an ALTUSER command with the KERBNAME keyword.
The **kerberos-principal-name** you define to RACF can consist of any character except the @ (X'7C') character. You can enter the name with or without single quotation marks, depending on the following:

- If parentheses, commas, blanks, or semicolons are entered as part of the name, the name must be enclosed in single quotation marks.
- If a single quotation mark is intended to be part of the name and the entire character string is enclosed in single quotation marks, you must use two single quotation marks together to represent each single quotation mark within the string.
- If the first character of the name is a single quotation mark, you must enter the string within single quotation marks, with two single quotation marks entered for that single quotation mark.

**Guideline:** Avoid using EBCDIC variant characters to prevent problems with different code pages.

Both uppercase and lowercase characters are accepted and maintained in the case in which they are entered. However, RACF does not ensure that a valid **kerberos-principal-name** has been specified.

A local **kerberos-principal-name** must **not** be qualified with a realm name when specified with the KERBNAME keyword. However, RACF verifies that the local principal name, when fully qualified with the name of the local realm:

```
/.../local_realm_name/principal_name
```

does not exceed 240 characters. For example,
- If the local realm name is
  
  X

  fully qualified local principal names are prefixed with
  
  /.../X/

  and are limited to a maximum of 233 characters.
- If the local realm name is
  
  KERB390.ENDICOTT.IBM.COM

  fully qualified local principal names are prefixed with
  
  /.../KERB390.ENDICOTT.IBM.COM/

  and are limited to a maximum of 210 characters.

This length verification requires that the REALM profile for the local realm KERBDFLT be defined and contain the name of the local realm, prior to the specification of local z/OS Network Authentication Service user principals. Otherwise, z/OS Network Authentication Service users will not be defined.

**Note:** Because of the relationship between realm names and local **kerberos-principal-names**, in which the length of a fully qualified name cannot exceed 240 characters, caution and planning must go into renaming the local realm since the combined length is only checked by RACF when a local...
ALTUSER

kerberos-principal-name is added or altered. Renaming the realm should be avoided as a result.

NOKERBNAME
Deletes the kerberos-principal-name. This invalidates the z/OS user ID’s z/OS Network Authentication Service account.

MAXTKTLFE | NOMAXTKTLFE

MAXTKTLFE(max-ticket-life)
Specifies the max-ticket-life in seconds. The value for MAXTKTLFE must be 1–2 147 483 647. Note that 0 is not a valid value.

If MAXTKTLFE is specified on the definition of a local z/OS Network Authentication Service principal, the z/OS Integrated Security Services Network Authentication Service takes the most restrictive of the value defined for the local principal and the value specified on the definition of the local realm (the KERBDFLT profile in the REALM class). Consequently, if the realm max-ticket-life is 24 hours, a principal cannot get a ticket with a longer lifetime even if the max-ticket-life is set to 48 hours. If this field is not specified for a local principal, or if NOMAXTKTLFE has been specified, the maximum lifetime for tickets created by this principal is determined from the definition of the local z/OS Network Authentication Service realm.

NOMAXTKTLFE
Deletes the max-ticket-life value for this local z/OS Network Authentication Service principal.

NOKERB
Deletes the user’s KERB segment. This user is no longer considered a principal by the z/OS Integrated Security Services Network Authentication Service.

LANGUAGE | NOLANGUAGE
Specifies to add, alter, or delete the user’s preferred national languages.

Specify LANGUAGE if this user is to have languages other than the ones established or defaulted by the LANGUAGE operand on the SETROPTS command, or the ones previously specified with the ADDUSER command.

LANGUAGE(PRIMARY(language) SECONDARY(language))
Specifies the user’s preferred national languages. Specify this operand if the user is to have languages other than the system-wide defaults (established by the LANGUAGE operand on the SETROPTS command).

- If this profile is for a TSO/E user who will establish an extended MCS console session, the languages you specify should be one of the languages specified on the LANGUAGE LANGCODE statements in the MMSLSTxx PARMLIB member. See your MVS system programmer for this information.
  For more information on TSO/E national language support, see z/OS TSO/E Customization
- If this profile is for a CICS user, see your CICS administrator for the languages supported by CICS on your system.
  For more information, see CICS RACF Security Guide

PRIMARY | NOPRIMARY
ALTUSER

PRIMARY(language)
Specifies the user’s new primary language.

NOPRIMARY
Deletes any primary language information from the user’s profile and returns the user to the installation’s default primary language.

SECONDARY | NOSECONDARY
SECONDARY(language)
Specifies the language to which the user’s secondary language is to be changed.

NOSECONDARY
Deletes any secondary language information from the user’s profile and returns the user to the installation’s default secondary language.

Notes:
1. For the primary and secondary languages, specify either the installation-defined name of a currently active language (a maximum of 24 characters) or one of the language codes (three characters in length) for a language installed on your system.
2. The language name can be a quoted or unquoted string.
3. The same language can be specified for with both PRIMARY and SECONDARY parameters.
4. If the MVS message service is not active, the PRIMARY and SECONDARY values must be a 3-character language code.

NOLANGUAGE
Deletes the user’s preferred national languages from the profile and returns that user to the installation defaults. LANGUAGE information no longer appears in LISTUSER output.

LNOTES | NOLNOTES
LNOTES
Specifies Lotus Notes for z/OS information for the user profile being changed.

SNAMENOSNAME
SNAME(short-name)
Specifies the Lotus Notes for z/OS short-name of the user being changed. The name should match the one stored in the Lotus Notes address book for this user, but this is not verified by the command.

The short-name you define to RACF can contain 1–64 characters. You can specify the following characters: uppercase and lowercase alphabetic characters (A–Z, and a–z), 0–9, & (X’50’), - (X’60’), . (X’4B’), _ (X’6D’), and blanks (X’40’).

If the short-name you specify contains any blanks, it must be enclosed in single quotation marks. The short-name is stripped of leading and trailing blanks.

The value specified for the short-name must be unique. Consequently, a list of users might not be specified on an ALTUSER command with the SNAME keyword.
ALTUSER

**NOSNAME**
Specifies that you want to delete the short-name from the LNOTES segment of the user’s profile.

**NOLNOTES**
Specifies that you want to delete the LNOTES segment from the user’s profile.

**MODEL | NOMODEL**

**MODEL(dsname)**
Specifies the name of a data set that RACF is to use as a model when new data set profiles are created that have userid as the high-level qualifier. For this operand to be effective, the MODEL(USER) option (specified on the SETROPTS command) must be active. If the ALTUSER command cannot find the dsname profile, it issues a warning message but places the model name in the user ID entry.

Note that RACF always prefixes dsname with the user ID.

Deletes the model profile name in the user’s profile.

**NAME(user-name)**
Specifies the user name to be associated with the user ID. You can use a maximum of 20 alphanumeric or non-alphanumeric characters. If the name you specify contains any blanks, it must be enclosed in single quotation marks.

Names longer than 20 characters are truncated to 20 characters when you enclose the name in quotation marks. However, if you specify a name longer than 20 characters without enclosing the name in quotation marks, you receive an error from the TSO parse routine.

If you omit the NAME operand, RACF uses a default of twenty # (X’7B’) characters (‘###’...’). Note, however, that the corresponding entry in a LISTUSER output is the word UNKNOWN.

**NDS | NONDS**

**NDS**
Specifies Novell Directory Services for OS/390 information for the user profile being changed.

**UNAME | NOUNAME**

**UNAME(user-name)**
Specifies the Novell Directory Services for OS/390 user-name of the user being changed. The user-name value should match the name stored in the Novell Directory Services for OS/390 directory for this user, but this is not verified by the command.

The user-name you define to RACF can contain 1–246 characters. However, the user-name cannot contain the following characters: * (X’5C’), + (X’4E’), | (X’4F’), ~ (X’7E’), , (X’5E’), " (X’27’), / (X’61’), : (X’3A’), ; (X’5E’), ¢ (X’4A’), and brackets [ and ] (X’AD’ and X’BD’).

If the user-name you specify contains any parentheses or blanks, it must be enclosed in single quotation marks. The user-name is stripped of leading and trailing blanks. If a single quotation mark is
ALTUSER

intended to be part of the *user-name*, use two single quotation
marks together for each single quotation mark within the string, and
enclose the entire string within single quotation marks.

The value specified for the *user-name* must be unique.
Consequently, a list of users cannot be specified on an ALTUSER
command with the UNAME keyword.

**NOUNAME**
Specifies that you want to delete the *user-name* from the NDS
segment of the user’s profile.

**NONDS**
Specifies that RACF delete the NDS segment from the user’s profile.

**NETVIEW | NONETVIEW**

**NETVIEW**
Specifies that this is a NetView operator that can enter any of the following
suboperands to add, update, or delete the information in the NETVIEW
segment.

You can control access to the entire NETVIEW segment or to individual
fields within the NETVIEW segment by using field-level access checking.
For more information, see z/OS Security Server RACF Security
Administrator’s Guide.

**CONSNAME | NOCONSNAME**

**CONSNAME**(console-name)
Specifies the default MCS console name identifier used for this
operator. This default console name is used when the operator
does not specify a console name using the NetView GETCONID
command.

The *console-name* value is a 1–8 character identifier whose validity
is checked by MVS processing when the operator tries to use it.
See z/OS MVS Planning: Operations for information on valid values
for a particular release.

**NOCONSNAME**
Deletes any default MCS console name previously specified for this
operator.

**CTL | NOCTL**

**CTL (GENERAL | GLOBAL | SPECIFIC)**
Specifies whether a security check is performed for this NetView
operator when they try to use a span or try to do a cross-domain
logon.

**GENERAL**
Specifies that a security done should be done as for SPECIFIC,
and, in addition, that the operator is allowed to access devices
that are not part of any span.

**GLOBAL**
Specifies that no security check is done.

**SPECIFIC**
Specifies that a security check is performed through
RACROUTE REQUEST=AUTH whenever this operator
attempts to use a span. It also specifies that any cross-domain
ALTUSER

Logon must be to a domain listed in the operator's NETVIEW segment with the DOMAINS keyword.

CTL(SPECIFIC) is the default.

NOCTL
NOCTL has the same effect as specifying CTL(SPECIFIC).

DOMAINS | NODOMAINS | ADDDOMAINS | DELDOMAINS

DOMAINS(domain-name ...)
Specifies the complete list of identifiers of NetView programs in another NetView domain where this operator can start a cross-domain session. The NetView program identifiers are coded on the NCCFID definition statement for the other domains, and represent the name given to that NetView program on the APPL statement.

Domain-name is a 1–5 character identifier. The characters can be alphabetic, numeric, or national.

ADDDOMAINS(domain-name ...)
Adds identifiers of NetView programs in another NetView domain where this operator can start a cross-domain session. The NetView program identifiers are coded on the NCCFID definition statement for the other domains, and represent the name given to that NetView program on the APPL statement.

The domain-name value is a 1–5 character identifier. The characters can be alphabetic, numeric, or national.

DELDOMAINS(domain-name ...)
Deletes specific identifiers of NetView programs in another NetView domain where this operator can start a cross-domain session. The NetView program identifiers are coded on the NCCFID definition statement for the other domains, and represent the name given to that NetView program on the APPL statement.

The domain-name value is a 1–5 character identifier. The characters can be alphabetic, numeric, or national.

NODOMAINS
Specifies that the operator cannot start any cross-domain sessions.

IC | NOIC

IC('command | command-list')
Specifies the command or command list (up to 255 characters) to be processed when the operator logs on to NetView.

If the command or command list you specify contains any commas, blanks, or other special characters that TSO/E requires to be quoted, it must be enclosed in single quotation marks.

NOIC
Deletes the command or command list to be processed at logon time for this operator. No command or command list is automatically processed when this operator logs on.

MSGRECVR | NOMSGRECVR

MSGRECVR (YES | NO)
Specifies whether this operator can receive unsolicited messages that are not routed to a specific NetView operator.
ALTUSER

YES     Specifies that the operator is to receive the messages.
NO      Specifies that the operator is not to receive the messages.

NOMSGRECVR
NOMSGRECVR has the same effect as specifying
MSGRECVR(NO).

NGMFADMN | NONGMFADMN

NGMFADMN (YES | NO)
Specifies whether a NetView operator has administrator authority to
the NetView Graphic Monitor Facility (NGMF).

YES     Specifies that the operator does have the authority.
NO      Specifies that the operator does not have the authority.

NONGMFADMN
NONGFMADMN has the same effect as specifying
NGFMADMN(NO).

NGMFVSPN | NONGMFVSPN

NGMFVSPN (view-span)
Reserved for future use by the NetView Graphic Monitor Facility

NONGMFVSPN
Reserved for future use by the NetView Graphic Monitor Facility

OPCLASS | NOOPCLASS | ADDOPCLASS | DELOPCLASS

OPCLASS(class ...)
Specifies the complete list of NetView scope classes for which the
operator has authority.

The class value is a number 1–2040 that specifies a NetView scope
class.

ADDOPCLASS(class ...)
Adds specific NetView scope classes to the operator’s current list of
classes.

The class value is a number 1–2040 that specifies a NetView scope
class.

DELOPCLASS(class ...)
Deletes specific NetView scope classes from the operator’s current
list of classes.

The class value is a number 1–2040 that specifies a NetView scope
class.

NOOPCLASS
Specifies that the operator is in no scope classes.

NONETVIEW
Specifies that RACF should delete the NETVIEW segment from the user’s
profile.

OIDCARD | NOOIDCARD

OIDCARD
Specifies that the user must supply an operator identification card when
logging onto the system. If you specify the OIDCARD operand, the system
prompts you to enter the user’s new operator identification card as part of
ALTUSER

the processing of the ALTUSER command. If you specify the OIDCARD operand in a job executing in the background or when you cannot be prompted in the foreground, the ALTUSER command fails.

NOOIDCARD
Specifies that the user is not required to supply an operator identification card.

If NOPASSWORD is specified or the user ID already has the NOPASSWORD attribute, specifying NOOIDCARD causes this user ID to become a protected user ID. Protected user IDs cannot be used to enter the system by any means that requires a password to be specified, such as TSO logon. If the user attempts to enter the system with a password, the attempt fails.

Protected user IDs can be used for the user IDs associated with the started tasks in ICHRIN03 or the STARTED class.

OMVS | NOOMVS

OMVS
Specifies z/OS UNIX information for the user profile being changed.

You can control access to the entire OMVS segment or to individual fields within the OMVS segment by using field-level access checking.

ASSIZEMAX | NOASSIZEMAX

ASSIZEMAX(address-space-size)
Specifies the RLIMIT_AS hard limit (maximum) resource value that processes receive when they are dubbed a process. The address-space-size you define to RACF is a numeric value from 10485760–2 147 483 647. ASSIZEMAX indicates the address space region size in bytes. The soft limit (current) resource value is obtained from MVS. If the soft limit value from MVS is greater than the address space size, the soft limit is used.

The value specified for ASSIZEMAX is also used when processes are initiated by a daemon process using an exec after setuid(). In this case, both the RLIMIT_AS hard limit and soft limit are set to the address-space-size value.

The value specified for ASSIZEMAX overrides any value provided by the MAXASSIZE parameter of BPXPRMxx. For more information, see z/OS UNIX System Services Planning.

NOASSIZEMAX
Specifies that you want to delete the address space size from the OMVS segment of the user’s profile. The value specified for MAXASSIZE in BPXPRMxx now applies to the user.

AUTOUID | UID | NOUID

Specifies whether RACF is to automatically assign an unused UID value to the user, if a specific UID value is to be assigned or if the user identifier from the OMVS segment of the user’s profile is to be deleted.

AUTOUID
Specifies that RACF is to automatically assign an unused UID value to the user. The UID value is derived from information obtained from the BPX.NEXT.USER profile in the FACILITY class. For more information on setting up BPX.NEXT.USER, see z/OS Security Server RACF Security Administrator's Guide.
ALTUSER

If you are using RRSF automatic command direction for the USER class, the command sent to other nodes will contain an explicit assignment of the UID value which was derived by RACF on the local node.

Rules:
- AUTOUID cannot be specified if more than one user ID is entered.
- The AUTOUID keyword is mutually exclusive with the SHARED keyword.
- If both UID and AUTOUID are specified, AUTOUID is ignored.
- If both NOUID and AUTOUID are specified, AUTOUID is ignored.
- Field-level access checking for the UID field applies when using AUTOUID.
- AUTOUID cannot be used to reassign a UID value when one already exists for the user. If AUTOUID is specified, but the user already has a UID assigned, one of two things will happen.
  - If the preexisting UID is unique to this user, this value will be identified in informational message IRR52177I, and the value will be left unchanged. If RRSF automatic command direction is in effect for the USER class, then the outbound ALTUSER command will be altered to contain the preexisting UID value in the OMVS UID keyword.
  - If the preexisting UID is not unique to this user, error message IRR52178I will be issued, and the command will fail. See IRR52178I for information on changing the user's existing UID value.

UID(user-identifier) [SHARED]

UID(user-identifier)

Specifies the user identifier. The UID is a numeric value from 0–2 147 483 647.

When assigning a UID to a user, you should make sure that the user’s default group has a GID. A user who has a UID and a current connect group that has a GID can use functions such as the TSO/E OMVS command and can access z/OS UNIX files based on the UID and GID values assigned.

Care should be taken in assigning 0 as the user identifier. UID 0 is considered a superuser. The superuser passes all z/OS UNIX security checks. Assigning a UID to a user ID that appears in the RACF started procedures table (ICHRIN03) should also be done with care. RACF defined started tasks that have the trusted or privileged attribute are considered superusers even if their UID is a value other than 0.

If the UID is not specified, the user is unable to become a z/OS UNIX user and a LISTUSER for that user ID shows NONE for the UID.

Notes:
1. If the security administrator has defined the SHARED.IDS profile in the UNIXPRIV class, the UID value must be unique. Use the SHARED keyword in addition to UID to assign a value that is already in use.
2. If SHARED.IDS is not defined, RACF does not require the UID to be unique. The same value can be assigned to multiple users but this is not recommended because individual user control would be lost. However, if you want a set of users to have exactly the same access to z/OS UNIX resources, you might decide to assign the same UID to more than one user.

3. The maximum number of user IDs that can share a UID or groups that can share a GID is 132 at 8 characters. More user IDs or groups are available using less than 8 characters. If the limit is met, you can combine user ID functions (for started tasks or daemons) to use physically less user IDs sharing the same UID. You may also use SUPERUSER granularity functionality to reduce the need for SUPERUSER (using UID 0) for as many user IDs as possible.

**SHARED**

If the security administrator has chosen to control the use of shared UIDs, this keyword must be used in addition to the UID keyword to specify the user identifier if it is already in use by at least one other user. The administrator controls shared UIDs by defining the SHARED.IDS profile in the UNIXPRIV class.

**Rules:**

- If the SHARED.IDS profile is not defined, SHARED is ignored.
- If SHARED is specified in the absence of UID, it is ignored.
- If the SHARED.IDS profile is defined and SHARED is specified, but the value specified with UID is not currently in use, SHARED is ignored and UNIXPRIV authority is not required.
- Field-level access checking for the UID field applies when using SHARED.
- The SHARED keyword is mutually exclusive with the AUTOUID keyword.

**NOUID**

Specifies that you want to delete the user identifier from the OMVS segment of the user’s profile.

If NOUID is specified, the user is unable to become a z/OS UNIX System Services user and a LISTUSER for that user ID shows NONE for the UID.

**CPUTIMEMAX | NOCPUTIMEMAX**

**CPUTIMEMAX**(cpu-time)  
Specifies the RLIMIT_CPU hard limit (maximum) resource value that the user’s z/OS UNIX processes receive when they are dubbed a process. The cpu-time you define to RACF is a numeric value from 7–2 147 483 647. RLIMIT_CPU indicates the cpu-time that a process is allowed to use, in seconds. The soft limit (current) is obtained from MVS. If the soft limit (current) resource value from MVS is greater than the cpu-time value, the soft limit is used.
ALTUSER

The value specified for CPUTIMEMAX is also used when processes are initiated by a daemon process using an exec after setuid(). In this case, both the RLIMIT_CPU hard and soft limits are set to the cpu-time value.

For processes running in, or forked from TSO or BATCH, the cpu-time value has no effect. For processes created by the rlogin command or other daemons, cpu-time is the time limit for the address space.

The value specified for CPUTIMEMAX overrides any value provided by the MAXCPUTIME parameter of BPXPRMxx. For more information, see z/OS UNIX System Services Planning.

NOCPUTIMEMAX

Specifies that you want to delete the CPU time from the OMVS segment of the user’s profile. The value specified for MAXCPUTIME in BPXPRMxx now applies to the user.

FILEPROCMAX | NOFILEPROCMAX

FILEPROCMAX(files-per-process)

Specifies the maximum number of files the user is allowed to have concurrently active or open. The files-per-process you define to RACF is a numeric value from 3 and 524287. FILEPROCMAX is the same as the OPEN_MAX variable defined in the POSIX standard.

FILEPROCMAX lets you limit the amount of system resources available to a user process. Select FILEPROCMAX by considering:

- For conformance to standards, set FILEPROCMAX to:
  - At least 16 to conform to the POSIX standard
  - At least 25 to conform to the FIPS standard

- The commonly recommended value is 256.

- A process can change its own value for the number of files it has active or open using the setrlimit() function. Only processes with appropriate privileges can increase their limits.

- The minimum value of 3 supports the standard files for a process: standard input, standard output, and standard error.

- The value needs to be larger than 3 to support z/OS UNIX shell users. If the value is too small, the z/OS UNIX shells might issue the message File descriptor not available.

The value specified for FILEPROCMAX overrides any value provided by the MAXFILEPROC parameter of BPXPRMxx. For more information, see z/OS UNIX System Services Planning.

NOFILEPROCMAX

Specifies that you want to delete the files per process from the OMVS segment of the user’s profile. The value specified for MAXFILEPROC in BPXPRMxx now applies to the user.

HOME | NOHOME

HOME(directory-pathname)

Specifies the z/OS UNIX directory pathname. This is the current working directory for the user’s process when the user enters the TSO/E command OMVS.
ALTUSER

When you define a directory pathname to RACF, it can contain 1–1023 characters. The directory pathname can consist of any characters and can be entered with or without single quotation marks. The following rules apply:

• If parentheses, commas, blanks, or semicolons are to be entered as part of the pathname, the character string must be enclosed in single quotation marks.
• If a single quotation mark is intended to be part of the pathname, use two single quotation marks together for each single quotation mark within the string, and enclose the entire string within single quotation marks.

Both uppercase and lowercase characters are accepted and maintained in the case in which they are entered. The fully qualified pathname should be specified. RACF does not ensure that a valid pathname has been specified. If you issue the ALTUSER command as a RACF operator command and you specify the pathname in lowercase, you must include the pathname within single quotations.

NOHOME

Specifies that you want to delete the initial directory pathname from the OMVS segment of the user’s profile.

If no value is specified for HOME in the OMVS segment, MVS sets the working directory for the user to / (the root directory).

MEMLIMIT | NOMEMLIMIT

MEMLIMIT(nonshared-memory-size)

Specifies the maximum number of bytes of nonshared memory that can be allocated by the user. The nonshared-memory-size value must be numeric 0–16777215, followed by the letter M, G, T, or P. The M, G, T or P letter indicates the multiplier to be used. The maximum value is 16383P.

<table>
<thead>
<tr>
<th>Decimal</th>
<th>Binary</th>
<th>Hexadecimal</th>
</tr>
</thead>
<tbody>
<tr>
<td>M—megabyte</td>
<td>1048576</td>
<td>2^20</td>
</tr>
<tr>
<td>G—gigabyte</td>
<td>1073741824</td>
<td>2^30</td>
</tr>
<tr>
<td>T—terabyte</td>
<td>1099511627776</td>
<td>2^40</td>
</tr>
<tr>
<td>P—petabyte</td>
<td>1125899906842624</td>
<td>2^50</td>
</tr>
</tbody>
</table>

The following are different MEMLIMIT(nonshared-memory-size) examples:

• MEMLIMIT(1M) indicates a nonshared-memory-size of 1048576 bytes.
• MEMLIMIT(1500M) indicates a nonshared-memory-size of 1572864000 bytes.
• MEMLIMIT(10G) indicates a nonshared-memory-size of 10737418240 bytes.

For more extensive information, see z/OS UNIX System Services Planning.

NOMEMLIMIT

Specifies that you want to delete the nonshared memory size from the OMVS segment of the user’s profile.
**MMAPAREAMAX | NOMMAPAREAMAX**

**MMAPAREAMAX** *(memory-map-size)*

Specifies the maximum amount of data space storage, in pages, that can be allocated by this user for memory mappings of z/OS UNIX files. Storage is not allocated until memory mappings are active. The value of *memory-map-size* must be 1–16777216.

Use of memory map services consumes a significant amount of system memory. For each page (4KB) that is memory mapped, 96 bytes of ESQA are consumed when a file is not shared with any other users. When a file is shared by multiple users, each user after the first causes 32 bytes of ESQA to be consumed for each shared page. The ESQA storage is consumed when the *mmap()* function is invoked by the application program.

The value specified for MMAPAREAMAX overrides any value provided by the MAXMMAPAREA parameter of BPXPRMxx. For more information, see [z/OS UNIX System Services Planning](#).

**NOMMAPAREAMAX**

Specifies that you want to delete the memory map size from the OMVS segment of the user's profile. The value specified for MAXMMAPAREA in BPXPRMxx now applies to the user.

**PROCUSERMAX | NOPROCUSERMAX**

**PROCUSERMAX** *(processes-per-UID)*

Specifies the maximum number of processes this user is allowed to have active at the same time, regardless of how the process became an z/OS UNIX process. The *processes-per-UID* you define to RACF is a numeric value from 3 and 32767. PROCUSERMAX is the same as the CHILD_MAX variable defined in the POSIX standard.

PROCUSERMAX allows you to limit user activity to optimize performance. Select PROCUSERMAX by considering:

- For conformance to standards, set PROCUSERMAX to:
  - At least 16 to conform to the POSIX standard
  - At least 25 to conform to the FIPS standard
- A user with a UID of 0 is not limited by the PROCUSERMAX value because a superuser might need to be capable of logging on and using z/OS UNIX services to solve a problem.
- A low PROCUSERMAX value limits the number of concurrent processes that the user can run. A low value also limits the user's consumption of processing time, virtual storage, and other system resources.
- Some daemons run without UID 0, and can create many address spaces. In these cases, it is necessary to set the limit high enough for the daemon associated with this user ID to run all of its processes.

Though not recommended, the same OMVS UID can be given to more than one user ID. If users share a UID, you need to define a greater number for PROCUSERMAX. An example is the user ID defined for the default OMVS segment, specified by the FACILITY class profile BPX.DEFAULT.USER.
ALTUSER

The value specified for PROCUSERMAX overrides any value provided by the MAXPROCUSER parameter of BPXPRMxx. For more information, see z/OS UNIX System Services Planning.

NOPROCUSERMAX
Specifies that you want to delete the processes per UID from the OMVS segment of the user’s profile. The value specified for MAXPROCUSER in BPXPRMxx now applies to the user.

SHMEMMAX | NOSHMEMMAX

SHMEMMAX(shared-memory-size)
Specifies the maximum number of bytes of shared memory that can be allocated by the user. The shared-memory-size value must be numeric 1–16777215, followed by the letter M, G, T, or P. The M, G, T, or P letter indicates the multiplier to be used. The maximum value is 16383P.

<table>
<thead>
<tr>
<th>Decimal</th>
<th>Binary</th>
<th>Hexadecimal</th>
</tr>
</thead>
<tbody>
<tr>
<td>M—megabyte</td>
<td>1048576</td>
<td>230</td>
</tr>
<tr>
<td>G—gigabyte</td>
<td>1073741824</td>
<td>230</td>
</tr>
<tr>
<td>T—terabyte</td>
<td>1099511627776</td>
<td>240</td>
</tr>
<tr>
<td>P—petabyte</td>
<td>1125899906842624</td>
<td>250</td>
</tr>
</tbody>
</table>

The following are different SHMEMMAX(shared-memory-size) examples:
- SHMEMMAX(1M) indicates a shared-memory-size of 1048576 bytes.
- SHMEMMAX(1500M) indicates a shared-memory-size of 1572864000 bytes.
- SHMEMMAX(10G) indicates a shared-memory-size of 10737418240 bytes.

The value specified for SHMEMMAX overrides any value provided by the IPCSHMMPAGES parameter of BPXPRMxx. For more information, see z/OS UNIX System Services Planning.

NOSHMEMMAX
Specifies that you want to delete the shared memory size from the OMVS segment of the user’s profile. The value specified for IPCSHMMPAGES in BPXPRMxx now applies to the user.

THREADSMAX | NOTHREADSMAX

THREADSMAX(threads-per-process)
Specifies the maximum number of threads, including those running, queued, and exited but not detached, that this user can have concurrently active. The threads-per-process you define to RACF is a numeric value from 0–100000. Specifying a value of 0 prevents applications run by this user from using the pthread_create service.

The value specified for THREADSMAX overrides any value provided by the MAXTHREADS parameter of BPXPRMxx. For more information, see z/OS UNIX System Services Planning.

NOTHREADSMAX
Specifies that you want to delete the threads per process from the OMVS segment of the user’s profile. The value specified for MAXTHREADS in BPXPRMxx now applies to the user.
PROGRAM | NOPROGRAM

PROGRAM(*program-name*)

Specifies the PROGRAM pathname (z/OS UNIX shell program). This is the first program started when the TSO/E command OMVS is entered or when a batch job is started using the BPXBATCH program.

When you define a PROGRAM pathname to RACF, it can contain 1–1023 characters. The PROGRAM pathname can consist of any characters and can be entered with or without single quotation marks. The following rules apply:

- If parentheses, commas, blanks, or semicolons are to be entered as part of the pathname, the character string must be enclosed in single quotation marks.
- If a single quotation mark is intended to be part of the pathname, use two single quotation marks together for each single quotation mark within the string, and enclose the entire string within single quotation marks.

Both uppercase and lowercase characters are accepted and maintained in the case in which they are entered. The fully qualified pathname should be specified. RACF does not ensure that a valid pathname has been specified. If you issue the ALTUSER command as a RACF operator command and you specify the pathname in lowercase, you must include the pathname within single quotations.

NOPROGRAM

Specifies that you want to delete the z/OS UNIX System Services program pathname from the OMVS segment of the user's profile.

If no value is specified for PROGRAM in the OMVS segment, MVS gives control to the default z/OS UNIX shell program when the user issues the TSO/E command OMVS or starts a batch job using the BPXBATCH program.

For more information about the default z/OS UNIX shell program supplied with z/OS UNIX System Services, see z/OS UNIX System Services Planning and z/OS UNIX System Services User's Guide.

NOOMVS

Specifies that RACF delete the OMVS segment from the user's profile.

OPERATIONS | NOOPERATIONS

OPERATIONS

Specifies that the user is to have authorization to do maintenance operations on all RACF-protected DASD data sets, tape volumes, and DASD volumes except those where the access list specifically limits the OPERATIONS user to an access authority that is less than the operation requires.

You establish the lower access authority for the OPERATIONS user with the PERMIT command. OPERATIONS on the ALTUSER command overrides NOOPERATIONS on the CONNECT command.

You must have the SPECIAL attribute to use the OPERATIONS operand.

NOOPERATIONS

Specifies that the user is not to have the OPERATIONS attribute.
ALTUSER

You must have the SPECIAL attribute to use the NOOPERATIONS operand.

OPERPARM | NOOPERPARM

OPERPARM

Specifies default information used when this user establishes an extended MCS console session.

You can control access to the entire OPERPARM segment or to individual fields within the OPERPARM segment by using field-level access checking. For more information, see [z/OS Security Server RACF Security Administrator's Guide](#).

For information on planning how to use OPERPARM segments, see [z/OS MVS Planning: Operations](#).

Notes:

1. You need not specify every suboperand in an OPERPARM segment. In general, if you omit a suboperand, the default is the same as the default in the CONSOLExx PARMLIB member, which can also be used to define consoles.

2. If you specify MSCOPE or ROUTCODE but do not specify a value for them, RACF uses MSCOPE(+ALL) and ROUTCODE(NONE) to update the corresponding fields in the user profile. These values appear in listings of the OPERPARM segment of the user profile.

3. If you omit the other suboperands, RACF does not update the corresponding fields in the user's profile, and no value appears in listings of the OPERPARM segment of the profile.

ALTGRP | NOALTGRP

ALTGRP(alternate-console-group)

Specifies the console group used in recovery. The variable alternate-console-group can contain 1–8 characters. Valid characters are 0–9, A–Z, # (X'7B'), $ (X'5B'), or @ (X'7C').

Restriction: Starting with z/OS Version 1 Release 8, console services ignores ALTGRP(alternate-console-group) when a session is established and it need not be specified.

NOALTGRP

Deletes alternate console group information from this profile.

AUTH | NOAUTH

AUTH

Specifies this console’s authority to issue operator commands.

If you omit this operand, RACF does not alter this field in the user’s profile. If this field has not been added to the user’s profile, an extended MCS console uses AUTH(INFO) when a session is established. The console can have the following authorities:

MASTER

Allows this console to act as a master console, which can issue all MVS operator commands. This authority can only be specified by itself.

ALL

Allows this console to issue system control commands,
input/output commands, console control commands, and informational commands. This authority can only be specified by itself.

INFO
Allows this console to issue informational commands. This authority can only be specified by itself.

CONS
Allows this console to issue console control and informational commands.

IO
Allows this console to issue input/output and informational commands.

SYS
Allows this console to issue system control commands and informational commands.

NOAUTH
Deletes the user's operator authorities from the profile. Console operator authority no longer appears in profile listings. However, AUTH(INFO) is used when an extended MCS console session is established.

AUTO | NOAUTO
AUTO(YES | NO)
Specifies whether the extended console can receive messages that have been automated by the Message Processing Facility (MPF) in the sysplex.

NOAUTO
Deletes this field from the user's profile. No AUTO information appears in profile listings. However, AUTO(NO) is used when an extended MCS console session is established.

CMDSYS | NOCMDSYS
CMDSYS(system-name | *)
Specifies the system to which commands from this console are to be sent.

If you omit this operand, RACF does not alter this field in the user's profile. If this field has not been added to the user's profile, an extended MCS console uses CMDSYS(*) when a session is established. The system-name value must be 1–8 characters. Valid characters are A–Z, 0–9, @ (X'7C'), # (X'7B'), and $ (X'5B'). If (*) is specified, commands are processed on the local system where the console is attached.

NOCMDSYS
Deletes any system-names from this profile. No CMDSYS information appears in profile listings. However, CMDSYS(*) is used when an extended MCS console session is established.

DOM | NODOM
DOM(NORMAL | ALL | NONE)
Specifies which delete operator message (DOM) requests this console can receive.
ALTUSER

If you omit this operand, RACF does not alter this field in the user’s profile. If this field has not been added to the user’s profile, an extended MCS console uses DOM(NORMAL) when a session is established.

NORMAL
The system queues all appropriate DOM requests to this console.

ALL
All systems in the sysplex queue DOM requests to this console.

NONE
No DOM requests are queued to this console.

NODOM
Deletes this field from the user’s profile. DOM information no longer appears in profile listings. However, DOM(NORMAL) is used when an extended MCS console session is established.

HC | NOHC
HC(YES | NO)
Specifies whether this console is to receive messages that are directed to hardcopy.

NOHC
Deletes this field from the user’s profile. z/OS console services will use HC(NO) when a session is established.

INTIDS | NOINTIDS
INTIDS(YES | NO)
Specifies whether this console is to receive messages directed to console ID zero (the internal console). Such messages are usually responses to internally issued commands.

NOINTIDS
Deletes this field from the user’s profile. z/OS console services will use INTIDS(NO) when a session is established.

KEY | NOKEY
KEY(searching-key)
Specifies a 1–8 character name that can be used to display information for all consoles with the specified key by using the MVS command DISPLAY CONSOLES,KEY. If specified, KEY can include A–Z, 0–9, # (X’7B’), $ (X’5B’), or @ (X’7C’).

If you omit this operand, RACF does not alter this field in the user’s profile. If this field has not been added to the user’s profile, an extended MCS console uses a KEY value of NONE when a session is established.

NOKEY
Deletes search key information from the user’s profile. Search key information no longer appears in profile listings. However, a KEY value of NONE is used when an extended MCS console session is established.

LEVEL | NOLEVEL
LEVEL(message-level)
Specifies the messages that this console is to receive.
ALTUSER

If you omit this operand, RACF does not alter this field in the user's profile. If this field has not been added to the user's profile, an extended MCS console uses LEVEL(ALL) when a session is established.

The message-level variable can be a list of R, I, CE, E, IN, NB or ALL. If you specify ALL, you cannot specify R, I, CE, E, or IN.

NB The console receives no broadcast messages.
ALL The console receives these messages: R, I, CE, E, IN.
R The console receives messages requiring an operator reply.
I The console receives immediate action messages.
CE The console receives critical eventual action messages.
E The console receives eventual action messages.
IN The console receives informational messages.

NOLEVEL
Deletes any defined message levels for this console from the profile. Message information no longer appears in profile listings. However, LEVEL(ALL) is used when an extended MCS console session is established.

LOGCMDRESP | NOLOGCMDRESP
LOGCMDRESP(SYSTEM | NO)
Specifies if command responses are to be logged.
If you omit this operand, RACF does not alter this field in the user's profile. If this field has not been added to the user's profile, an extended MCS console uses LOGCMDRESP(SYSTEM) when a session is established.

SYSTEM
Specifies that command responses are logged in the hardcopy log.

NO
Specifies that command responses are not logged.

NOLOGCMDRESP
Deletes the value for LOGCMDRESP from the profile. Command response logging information no longer appears in profile listings. However, LOGCMDRESP(SYSTEM) is used when an extended MCS console session is established.

MFORM | NOMFORM
MFORM(message-format)
Specifies the format in which messages are displayed at the console.
If you omit this operand, RACF does not alter this field in the user's profile. If this field has not been added to the user's profile, an extended MCS console uses MFORM(M) when a session is established.

The message-format variable can be a combination of T, S, J, M, and X:
J Messages are displayed with a job ID or name.
**M** The message text is displayed.

**S** Messages are displayed with the name of the originating system.

**T** Messages are displayed with a time stamp.

**X** Messages that are flagged as exempt from job name and system name formatting are ignored.

**NOMFORM**
Delete the values for MFORM from the profile and causes message text to be displayed (MFORM(M)) when an extended MCS console session is established.

**MIGID | NOMIGID**

**MIGID(YES | NO)**
Specifies whether a 1-byte migration ID is to be assigned to this console or not. The migration ID allows command processors that use a 1-byte console ID to direct command responses to this console.

**Restriction:** Starting with z/OS Version 1 Release 7, console services ignores MIGID(YES | NO) when a session is established and it need not be specified.

**NOMIGID**
Deletes this segment from the profile. Migration identification information no longer appears in profile listings. However, MIGID(NO) is assigned when an extended MCS console session is established.

**MONITOR | NOMONITOR**

**MONITOR(events)**
Specifies which information should be displayed when monitoring jobs, TSO sessions, or data set status.

If you omit this operand, RACF does not alter this field in the user’s profile. If this field has not been added to the user’s profile, an extended MCS console uses MONITOR(JOBNAME SESS) when a session is established. The variable events can be a list of the following:

**JOBNAMES | JOBNAMEST**
Displays information about the start and end of each job.
JOBNAMES omits the times of job start and job end.
JOBNAMEST displays the times of job start and job end.

**SESS | SESST**
Displays information about the start and end of each TSO session. SESS omits the times of session start and session end. SESST displays the times of session start and session end.

**STATUS**
Specifies that the information displayed when a data set is freed or unallocated should include the data set status.

**NOMONITOR**
Deletes job monitor information from the user’s profile. Information
from this field no longer appears in profile listings. However, MONITOR(JOBNAMES SESS) is used when an extended MCS console session is established.

**MSCOPE | ADDMSCOPE | DELMSCOPE | NOMSCOPE**

**MSCOPE(system-name ... | * | *ALL)**
Specifies the systems from which this console can receive messages that are not directed to a specific console.

If you omit this operand, RACF does not alter this field in the user’s profile. If this field has not been added to the user’s profile, an extended MCS console uses MSCOPE(*ALL) when a session is established. If you specify MSCOPE but omit a value, RACF uses MSCOPE(*ALL) as the default to update this field in the user’s profile. *ALL appears in listings of the OPERPARM segment of the user’s profile.

*system-name ...* is a list of one or more system names, where a system name can be any combination of A–Z, 0–9, # (X’7B’), $ (X’5B’), or @ (X’7C’).

* Is the system on which the console is currently active.

*ALL Means all systems.

**ADDMSCOPE(system-name ...)**
Adds the specified system names to the existing list of systems from which this console can receive messages that are not directed to a specific console.

**DELMSCOPE(system-name ...)**
Deletes the specified system names from the existing list of systems from which this console can receive messages that are not directed to a specific console.

**NOMSCOPE**
Deletes any system name information from the user’s profile. Message reception information no longer appears in profile listings. However, MSCOPE(*ALL) is used when an extended MCS console session is established.

**ROUTCODE | NOROUTCODE**

**ROUTCODE(ALL | NONE | routing-codes)**
Specifies the routing codes of messages this operator is to receive.

If you omit this operand, RACF does not alter this field in the user’s profile. If this field has not been added to the user’s profile, an extended MCS console uses ROUTCODE(NONE) when a session is established. If you specify ROUTCODE but omit a value, RACF uses ROUTCODE(NONE) to update this field in the user’s profile. NONE appears in listings of the OPERPARM segment of the user’s profile.

The routing code information can be one of the following:

**ALL** Means all routing codes.

**NONE** Means no routing codes.

**routing-codes** Specifies one or more routing codes or sequences.
of routing codes. The routing codes can be a list of $n$ and $n_1:n_2$, where $n$, $n_1$, and $n_2$ are integers 1–128, and $n_1:n_2$ represents a range of routing codes from $n_1$ (low) to $n_2$ (high).

NOROUTCODE
Deletes routing code information from the user’s profile. Routing code information no longer appears in profile listings. However, ROUTCODE(NONE) is used when an extended MCS console session is established.

STORAGE | NOSTORAGE
STORAGE(amount)
Specifies the amount of storage in the TSO/E user’s address space that can be used for message queuing to this console.

If you omit this operand, RACF does not alter this field in the user’s profile. If this field has not been added to the user’s profile, an extended MCS console uses STORAGE(1) when a session is established. A value of 0 will appear in listings of the user’s profile to indicate that no value was specified. The variable amount must be a value from 1–2000.

NOSTORAGE
Deletes this field from the profile. A value of 0 appears in listings of the user’s profile to indicate that no value was specified. However, STORAGE(1) is used when an extended MCS console session is established.

UD | NOUD
UD(YES | NO)
Specifies whether this console is to receive undelivered messages. If you do not specify this operand, RACF does not alter the user’s profile.

Restriction: Starting with z/OS Version 1 Release 8, console services ignores UD(YES | NO) when a session is established and it need not be specified.

NOUD
Deletes the field from the profile. Undelivered message information no longer appears in profile listings. However, UD(NO) is used when an extended MCS console session is established.

UNKNIDS | NOUNKNIDS
UNKNIDS(YES | NO)
Specifies whether this console is to receive messages directed to unknown console IDs. Unknown consoles are typically one-byte console IDs that the system cannot unambiguously resolve.

NOUNKNIDS
Deletes this field from the user’s profile. z/OS console services will use UNKNIDS(NO) when a session is established.

NOOPERPARM
Specifies that the OPERPARM segment is to be deleted. Operator information no longer appears in LISTUSER output.
**ALTUSER**

**OVM | NOOVM**
Specifies OpenExtensions VM information for the user profile being changed. Information is stored in the OVM segment of the user's profile.

You can control access to an entire OVM segment or to individual fields within the OVM segment by using field level access checking.

**FSROOT | NOFSROOT**

**FSROOT**(\textit{file-system-root})
Specifies the pathname for the file system root.

When you define the FSROOT pathname to RACF, it can contain 1–1023 characters, consist of any character, and be entered with or without single quotation marks. The following rules apply:

- If the pathname contains parentheses, commas, blanks, or semicolons, enclose the character string in single quotation marks. For example, if the pathname is (123), you must enter \texttt{FSROOT(' (123) ')}.  
- If a single quotation mark is intended to be part of the pathname, use two single quotation marks together for each single quotation mark within the string, and enclose the entire string within single quotation marks.

When entering the ALTUSER command, both uppercase and lowercase characters are accepted and maintained in the case in which they are entered. You should specify the fully qualified pathname because RACF does not ensure that a valid pathname has been specified.

**NOFSROOT**
Specifies that you want to delete the FSROOT pathname from the OVM segment of the user's profile.

If you do not specify a value for FSROOT in the OVM segment, VM uses the value specified in the CP directory. If no value is specified in the CP directory, issue the OPENVM MOUNT command to mount the appropriate file system.

**HOME | NOHOME**

**HOME**(\textit{initial-directory-name})
Specifies the initial directory pathname. The initial directory is part of the file system and is the current working directory for the user's process when the user enters the OPENVM SHELL command.

When you define a HOME directory name to RACF, the name can contain 1–1023 characters, consist of any character, and be entered with or without single quotation marks. The following rules apply:

- If the pathname contains parentheses, commas, blanks, or semicolons, enclose the character string in single quotation marks. For example, if the pathname is (123), you must enter \texttt{HOME(' (123) ')}.  
- If a single quotation mark is intended to be part of the pathname, use two single quotation marks together for each single quotation mark within the string, and enclose the entire string within single quotation marks.

When entering the ALTUSER command, both uppercase and lowercase characters are accepted and maintained in the case in which they are entered.
ALTUSER

entered. You should specify the fully qualified pathname because RACF does not ensure that a valid pathname has been specified.

NOHOME

Specifies that you want to delete the initial directory pathname from the OVM segment of the user’s profile.

If no value is specified for HOME in the OVM segment, VM uses the value specified in the CP directory. If no value is specified in the CP directory, VM sets the working directory for the user to / (the root directory).

PROGRAM | NOPROGRAM

PROGRAM(program-name)

Specifies the PROGRAM pathname (z/OS UNIX shell program). This is the first program started when the OPENVM SHELL command is entered.

When you define a PROGRAM pathname to RACF, it can contain 1–1023 characters, consist of any character, and be entered with or without single quotation marks. The following rules apply:
• If the pathname contains parentheses, commas, blanks, or semicolons, enclose the character string in single quotation marks. For example, if the pathname is (123), you must enter PROGRAM(‘(123)’).
• If a single quotation mark is intended to be part of the pathname, use two single quotation marks together for each single quotation mark within the string, and enclose the entire string within single quotation marks.

When entering the ALTUSER command, both uppercase and lowercase characters are accepted and maintained in the case in which they are entered. Specify the fully qualified pathname. RACF does not ensure that a valid pathname is specified.

NOPROGRAM

Specifies that you want to delete the PROGRAM pathname from the OVM segment of the user’s profile.

If no value is specified for PROGRAM in the OVM segment, VM uses the value specified in the CP directory. If no value is specified in the CP directory, VM gives control to the default z/OS UNIX shell program (/bin/sh) when a user issues the OPENVM SHELL command.

UID | NOUID

UID(user-identifier)

Specifies the user identifier. The UID is a numeric value from 0–2 147 483 647.

Care should be taken in assigning 0 as the user identifier. UID 0 is considered a superuser, and a superuser passes all OpenExtensions VM security checks.

Notes:
1. RACF does not require the UID to be unique. You can assign the same value to multiple users, but this is not recommended because individual user control is lost. However, if you want a set of users to have exactly the same access to the OpenExtensions VM resources, you can assign the same UID to more than one user.
2. Exercise caution when changing the UID for a user.
   - The file-system might contain files that were created by the user, and thus contain the old UID as the file owner UID. Depending on the permission bits associated with the file, the user will probably lose access to those files.
   - If files already exist with an owner UID equal to the user’s new UID value, the user will probably gain access to these files.
   - If another user is subsequently added with the old value as its UID, then the user might have access to the old files.
   - If you have an EXEC.uid profile in the VMPOSIX class for the old UID value, make sure you delete this profile and create another to reflect the new value.

NOUID
Specifies that you want to delete the user identifier from the OVM segment of the user’s profile.

If NOUID is specified, the user is assigned the default UID of 4294967295 (X'FFFFFFFF') and a LISTUSER for that user ID shows NONE for the UID.

NOOVM
Specifies that RACF delete the OVM segment from the user’s profile.

OWNER(userid or group-name)
Specifies a RACF-defined user or group to be assigned as the new owner of the user’s profile.

PASSWORD | NOPASSWORD

PASSWORD[(password)]
Specifies the user’s logon password. Use this command to specify a password for a user who has forgotten his/her password. Unless the NOEXPIRED operand is also specified, this password is set expired, thus requiring the user to change the password at next logon or job start. Note that the password syntax rules your installation defines using SETROPTS PASSWORD do not apply to this password unless the NOEXPIRED operand is also included. (See the NOEXPIRED operand.)

If you specify PASSWORD without a value, the password is defaulted to the user’s default group name. If you specify PASSWORD without a value and specify DFLTGRP, the default password is the user’s old default group name.

Notes:
1. For z/OS Integrated Security Services Network Authentication Service support, this means the key is not generated for the default group. However, the default group continues to be used as the RACF password.
2. If the installation is maintaining user password history, the password that was in effect prior to issuing this command is stored as part of this history.
3. When the installation specifies a minimum change interval, RACF checks the number of days between password changes to ensure the minimum required days have elapsed each time users change their own passwords. RACF also checks the days when users change passwords using their IRR.PASSWORD.RESET authority unless the command issuer has CONTROL authority or higher.
NOPASSWORD
Specifies that the user does not need to supply an initial logon password when first entering the system if OIDCARD is also specified. If NOOIDCARD is specified, or the user ID has the NOOIDCARD attribute and you specify NOPASSWORD, you change the status of the user ID to protected. Protected user IDs cannot be used to enter the system by any means that requires a password to be specified, such as a TSO logon, CICS signon, batch job that specifies a password on the JOB statement. Therefore, user IDs that you assign to z/OS UNIX, UNIX daemons, started procedures, applications, servers or subsystems can be protected from being revoked when an incorrect password is entered. If the user attempts to enter the system with a password, the attempt fails. Note that the protected user ID is not revoked due to the failed password attempts even if the SETROPTS PASSWORD(REVOKE) option is in effect.

Determine which user IDs you wish to protect, ensuring that these user IDs will not be used in any circumstance where a password must be supplied. A protected user will have the PROTECTED attribute displayed in the output of the LISTUSER command. Protected users can be associated with started procedures defined in the STARTED class (preferred method) or in the started procedures table (ICHRIN03).

Note: z/OS Integrated Security Services Network Authentication Service information such as a local kerberos-principal-name must not be defined for protected user IDs, and these user IDs must not be used for z/OS Network Authentication Service authentication, since these authentication failures can result in user revocation.

PHRASE | NOPHRASE

PHRASE('password-phrase')
Specifies the user's password phrase. The password phrase you define is a text string of up to 100 characters and must be enclosed in single quotation marks. The password phrase is set expired unless NOEXPIRED is also specified.

When the new-password-phrase exit (ICHPWX11) is present and allows it, the password phrase can be 9–100 characters. When ICHPWX11 is not present, the password phrase must be 14–100 characters. Contact your system programmer to find out if your installation uses the new-password-phrase exit (ICHPWX11) or see z/OS Security Server RACF System Programmer’s Guide for programming details.

Every user that you assign a password phrase must have a password. If you attempt to remove the password from a user with a password phrase, or add a password phrase for a user with no password, the PHRASE operand is ignored and an error message issued.

The following syntax rules apply to all password phrases. You cannot alter these syntax rules but you can specify additional syntax rules if your installation tailors the new-password-phrase exit (ICHPWX11).

Syntax rules for password phrases:
• Maximum length: 100 characters
• Minimum length:
  – 9 characters, when ICHPWX11 is present and allows the new value
  – 14 characters, when ICHPWX11 is not present
• Must not contain the user ID (as sequential uppercase or sequential lowercase characters)
ALTUSER

- Must contain at least 2 alphabetic characters (A–Z, a–z)
- Must contain at least 2 non-alphabetic characters (numerics, punctuation, or special characters)
- Must not contain more than 2 consecutive characters that are identical
- If a single quotation mark is intended to be part of the password phrase, you must use two single quotation marks together for each single quotation mark.

If the new-password-phrase exit (ICHPWX11) is present, it can reject the specified password phrase. RACF rejects password phrases shorter than 14 characters unless ICHPWXX11 is present and allows the new value.

If the specified password phrase is accepted, it is made the user’s current password phrase and, when SETROPTS PASSWORD(HISTORY) is in effect, it is added to the user’s password phrase history.

If you enter PHRASE without a password-phrase value, you are prompted for a value unless your TSO session is in NOPROMPT mode.

When the installation specifies a minimum change interval, RACF checks the number of days between password phrase changes to ensure the minimum required days have elapsed each time users change their own password phrases. RACF also checks the days when users change password phrases using the IRR.PASSWORD.RESET authority unless the command issuer has CONTROL authority or higher.

NOPHRASE

Specifies that the user cannot use a password phrase for authentication. If a password phrase was previously set, the password phrase is cleared. The date of the last password phrase change is also cleared from the user’s profile.

PROXY | NOPROXY

PROXY

Specifies information which the z/OS LDAP server will use when acting as a proxy on behalf of a requester. The R_proxyserv (IRRSPY00) SAF callable service will attempt to retrieve this information when it is not explicitly supplied with the invocation parameters. Applications or other services which use the R_proxyserv callable service, such as IBM Policy Director Authorization Services for z/OS and OS/390, may instruct their invokers to define PROXY segment information.

LDAPHOST | NOLDAPHOST

LDAPHOST(ldap_url)

Specifies the URL of the LDAP server which the z/OS LDAP server will contact when acting as a proxy on behalf of a requester. An LDAP URL has a format such as ldap://123.45.6:389 or ldaps://123.45.6:636, where ldaps indicates that an SSL connection is desired for a higher level of security. LDAP will also allow you to specify the host name portion of the URL using either the text form (BIGHOST:POK.IBM.COM) or the dotted decimal address (123.45.6). The port number is appended to the host name, separated by a colon (X'7A').

For more information about LDAP URLs and how to enable LDAP servers for SSL connections, see IBM Tivoli Directory Server Administration and Use for z/OS or z/OS Integrated Security Services LDAP Server Administration and Use.
ALTUSER

The LDAP URL that you define to RACF can consist of 10–1023 characters. A valid URL must start with either ldap:// or ldaps://. RACF will allow any characters to be entered for the remaining portion of the URL, but you should ensure that the URL conforms to TCP/IP conventions. For example, parentheses, commas, blanks, semicolons, and single quotation marks are not typically allowed in a host name. The LDAP URL can be entered with or without single quotation marks, however, in both cases, it will be translated to uppercase.

RACF does not ensure that a valid LDAP URL has been specified.

NOLDAPHOST

Deletes the URL of the LDAP server which the z/OS LDAP server will contact when acting as a proxy on behalf of a requester.

BINDDN | NOBINDDN

BINDDN(bind_distinguished_name)

Specifies the distinguished name (DN) which the z/OS LDAP server will use when acting as a proxy on behalf of a requester. This DN will be used in conjunction with the BIND password, if the z/OS LDAP server needs to supply an administrator or user identity to BIND with another LDAP server. A DN is made up of attribute value pairs, separated by commas. For example:

cn=Ben Gray,ou=editing,o=New York Times,c=US
cn=Lucille White,ou=editing,o=New York Times,c=US
cn=Tom Brown,ou=reporting,o=New York Times,c=US

When you define a BIND DN to RACF, it can contain 1–1023 characters. The BIND DN can consist of any characters and can be entered with or without single quotation marks. The following rules apply:

• If parentheses, commas, blanks, or semicolons are to be entered as part of the BIND DN, the character string must be enclosed in single quotation marks.

• If a single quotation mark is intended to be part of the BIND DN, use two single quotation marks together for each single quotation mark within the string, and enclose the entire string within single quotation marks.

Both uppercase and lowercase characters are accepted and maintained in the case in which they are entered. For more information about LDAP distinguished names, see IBM Tivoli Directory Server Administration and Use for z/OS or z/OS Integrated Security Services LDAP Server Administration and Use.

If you issue the ALTUSER command as a RACF operator command and you specify the BIND DN in lowercase, you must include the BIND DN within single quotations.

RACF does not ensure that a valid BIND DN has been specified.

NOBINDDN

Deletes the distinguished name (DN) used by the z/OS LDAP server when acting as a proxy on behalf of a requester.

BINDPW | NOBINDPW
ALTUSER

BINDPW
Specifies the password which the z/OS LDAP server will use when acting as a proxy on behalf of a requester.

When you define a BIND password to RACF, it can contain 1–128 characters. The BIND password can consist of any characters (see exception below) and can be entered with or without single quotation marks. The following rules apply:

- The BIND password cannot start with a left brace { character (X'8B').
- If parentheses, commas, blanks, or semicolons are to be entered as part of the BIND password, the character string must be enclosed in single quotation marks.
- If a single quotation mark is intended to be part of the BIND password, use two single quotation marks together for each single quotation mark within the string, and enclose the entire string within single quotation marks.

Both uppercase and lowercase characters are accepted and maintained in the case in which they are entered. For more information about LDAP passwords, see IBM Tivoli Directory Server Administration and Use for z/OS or z/OS Integrated Security Services LDAP Server Administration and Use.

If you issue the ALTUSER command as a RACF operator command and you specify the BIND password in lowercase, you must include the BIND password within single quotations.

RACF does not ensure that a valid BIND password has been specified.

Attention:
- When the command is issued from ISPF, the TSO command buffer (including possible BINDPW password data) is written to the ISPLOG data set. As a result, you should not issue this command from ISPF or you must control the ISPLOG data set carefully.
- When the command is issued as a RACF operator command, the command and the possible BINDPW password data is written to the system log. Therefore, use of ALTUSER as a RACF operator command should either be controlled or you should issue the command as a TSO command.

NOBINDPW
Deletes the password used by the z/OS LDAP server when acting as a proxy on behalf of a requester.

NOPROXY
Deletes LDAP proxy information.

RESTRIC TED | NORESTRICTED

RESTRIC TED
Specifies that global access checking is bypassed when resource access checking is performed for the user, and neither ID(+) on the access list nor the UACC will allow access. The RESTRICTED.FILESYS.ACCESS profile in the UNIXPRIV class can also be used to bypass the z/OS UNIX other permission bits during file access checking for RESTRICTED users.
ALTUSER

Note: If your installation has profiles defined in the PROGRAM class, and the user ID with the RESTRICTED attribute needs to load programs covered by one or more of these profiles, the user ID must be put on the access list with EXECUTE or READ authority.

NORESTRICTED
Specifies that the user does not have the RESTRICTED attribute and access checking is performed the standard way including global access checking, ID(*), the UACC, and the z/OS UNIX 'other' permission bits as appropriate.

RESUME | NORESUME

RESUME[(date)]
Specifies that the user is to be allowed to access the system again. You normally use RESUME to restore access to the system that has been prevented by a prior REVOKE.

If you specify a date, RACF prevents the user from accessing the system until the date you specify. The date must be a future date; if it is not, you are prompted to provide a future date.

Between the time you specify the RESUME and the time the RESUME takes effect, the RESUME is called a pending resumption (or a pending RESUME).

You specify a date in the form mm/dd/yy, and you need not specify leading zeros; specifying 9/1/06 is the same as specifying 09/01/06. RACF interprets dates as 20yy when yy is less than 71, and 19yy when yy is 71 or higher. So, 09/01/94 would be in the year 1994, and 09/01/14 would be in the year 2014.

If you specify RESUME without a date, the RESUME takes effect immediately.

When no REVOKE or pending REVOKE is in effect for the user, RACF ignores the RESUME operand.

Notes:

1. If you use the ALTUSER command to issue a REVOKE for a user, you must use the ALTUSER command to issue the corresponding RESUME. Issuing RESUME on the CONNECT command does not restore access revoked on the ALTUSER command.

2. If you specify both REVOKE(date) and RESUME(date), RACF acts on them in date order. For example, if you specify RESUME(8/19/06) and REVOKE(8/5/06), RACF prevents the user from accessing the system from August 5, 2006, to August 18, 2006. On August 19, the user can again access the system.

   If a user is already revoked and you specify RESUME(8/5/06) and REVOKE(8/19/06), RACF allows the user to access the system from August 5, 2006, to August 18, 2006. On August 19, RACF prevents the user from accessing the system.

3. If RACF detects a conflict between REVOKE and RESUME (for example, you specify both without a date), RACF uses REVOKE.

4. To clear the RESUME date field, specify NORESUME.

5. To successfully resume a user whose revoke date has passed, you must specify NOREVOKE to clear the revoke date as well as specifying the RESUME keyword.
6. Downlevel systems sharing the RACF database should not be affected by the changes to REVOKE and RESUME processing. A user who is considered revoked on a z/OS V1R7 system should also be considered revoked on a downlevel system.

NORESUME

Specifies that RACF is to clear the user's RESUME date field. You can use the NORESUME option to cancel the pending resumption (of a user's ID) that resulted from a previous ALTUSER command specified with RESUME(date).

REVOKE | NOREVOKE

REVOKE[(date)]

Specifies that RACF is to prevent the user from accessing the system. The user's profile is not deleted from the RACF database, and the user's data sets are not deleted from the RACF data set.

If you specify the date, RACF prevents the user from accessing the system, starting on the date you specify. The date must be a future date; if it is not, you are prompted to provide a future date.

Between the time you specify the REVOKE and the time the REVOKE takes effect, the REVOKE is called a pending revocation (or a pending REVOKE).

You specify a date in the form mm/dd/yy, and you need not specify leading zeros; specifying 9/1/06 is the same as specifying 09/01/06. RACF interprets dates as 20yy when yy is less than 71, and 19yy when yy is 71 or higher. So, 09/01/94 would be in the year 1994, and 09/01/14 would be in the year 2014.

When you specify REVOKE without a date, the following conditions apply:

• The REVOKE takes effect the next time the user tries to log onto the system.

• Any pending RESUME date remains in effect unless you also specify NORESUME.

Important: To permanently revoke system access, specify both REVOKE and NORESUME.

When a REVOKE is already in effect for the user, RACF ignores the REVOKE operand and issues a message.

Notes:

1. Specifying REVOKE on the ALTUSER command overrides RESUME on the CONNECT command.

2. If you specify both REVOKE(date) and RESUME(date), RACF acts on them in date order. For example, if you specify RESUME(8/19/06) and REVOKE(8/5/06), RACF prevents the user from accessing the system from August 5, 2006, to August 18, 2006. On August 19, the user can again access the system.

   If a user is already revoked and you specify RESUME(8/5/06) and REVOKE(8/19/06), RACF allows the user to access the system from August 5, 2006, to August 18, 2006. On August 19, RACF prevents the user from accessing the system.

3. If RACF detects a conflict between REVOKE and RESUME (for example, you specify both without a date), RACF uses REVOKE.

4. To clear the REVOKE date field, specify NOREVOKE.
5. Downlevel systems sharing the RACF database should not be affected by the changes to REVOKE and RESUME processing. A user who is considered revoked on a z/OS V1R7 system should also be considered revoked on a downlevel system.

**NOREVOKE**

Specifies that RACF is to clear the user’s REVOKE date field. You can use the NOREVOKE option to cancel the pending revocation (of a user’s ID) that resulted from a previous ALTUSER command specified with REVOKE(date).

To successfully resume a user whose revoke date has passed, you must specify NOREVOKE to clear the revoke date as well as specifying the RESUME keyword.

The NOREVOKE option does not resume the user ID after it was revoked by the ALTUSER REVOKE command or the user’s excessive attempts to use incorrect passwords or password phrases.

**SECLABEL | NOSECLABEL**

**SECLABEL(seclabel-name)**

Specifies the user’s default security label where seclabel-name is an installation-defined security label that represents an association between a particular security level and a set of zero or more security categories.

A security label corresponds to a particular security level (such as CONFIDENTIAL) with a set of zero or more security categories (such as PAYROLL or PERSONNEL).

When no profile in the SECLABEL class exists for seclabel-name, an error message is issued and the user’s security label is not changed.

**NOSECLABEL**

Specifies that the ALTUSER command is to delete any security label contained in the user profile.

**SECLEVEL | NOSECLEVEL**

**SECLEVEL(seclevel-name)**

Specifies the user’s security level, where seclevel-name is an installation-defined name that must be a member of the SECLEVEL profile in the SECDATA class. The security level name that you specify corresponds to the number of the minimum security level that a user must have to access the resource.

When you specify SECLEVEL and the SECDATA class is active, RACF adds security level access checking to its other authorization checking. If global access checking does not grant access, RACF compares the security level allowed in the user profile with the security level required in the resource profile. If the security level in the user profile is less than the security level in the resource profile, RACF denies the access. If the security level in the user profile is equal to or greater than the security level in the resource profile, RACF continues with other authorization checking.

**Note:** RACF does not perform security level checking for a started task or user that has the RACF privileged or trusted attribute. The RACF privileged or trusted attribute can be assigned to a started task through the RACF started procedures table or STARTED class, or to other users by installation-supplied RACF exits.
ALTUSER

When the SECDATA class is not active, RACF ignores this operand. When
the SECLEVEL profile does not include a member for selevel-name, you
are prompted to provide a valid security level name.

NOSECLEVEL
Specifies that the ALTUSER command is to delete any security level
contained in the user profile. The user no longer has access to any
resource that requires a requester to have a certain security level.

SPECIAL | NOSPECIAL
SPECIAL
Specifies that the user is to be allowed to issue all RACF commands with
all operands except the operands that require the AUDITOR attribute.
SPECIAL specified on the ALTUSER command overrides NOSPECIAL
specified on the CONNECT command.

You must have the SPECIAL attribute to use the SPECIAL operand.

NOSPECIAL
Specifies that the user no longer has the SPECIAL attribute.

You must have the SPECIAL attribute to use the NOSPECIAL operand.

TSO | NOTSO
TSO
Specifies that when you change the profile of a TSO user, you can enter
any of the following suboperands to add or change default TSO logon
information for that user. Each suboperand defines information that RACF
stores in a field within the TSO segment of the user’s profile.

You can control access to an entire TSO segment or to individual fields
within the TSO segment by using field-level access checking. For more
information, see z/OS Security Server RACF Security Administrator’s Guide.

ACCTNUM | NOACCTNUM
ACCTNUM(account-number)
Specifies the user’s default TSO account number when logging on
from the TSO/E logon panel. The account number you specify must be
defined as a profile in the ACCTNUM general resource class,
and the user must be granted READ access to the profile.
Otherwise, the user cannot log on to TSO using the specified
account number.

Account numbers can consist of any characters, and can be
entered with or without single quotation marks. The following rules
apply:
• If parentheses, commas, blanks, and semicolons are to be
taken as part of the account number, the character string must
be enclosed in single quotation marks. For example, if the
account number is (123), you must enter ACCTNUM(‘(123)’).
• If a single quotation mark is intended to be part of the account
number, use two single quotation marks together for each single
quotation mark within the string, and enclose the entire string
within single quotation marks.

A user can change an account number, or specify an account
number if one has not been specified, using the TSO/E logon panel.
RACF checks the user’s authorization to the specified account
number. If the user is authorized to use the account number, RACF
ALTUSER

stores the account number in the TSO segment of the user’s profile, and TSO/E uses it as a default value the next time the user logs on to TSO/E. Otherwise, RACF denies the use of the account number.

**Note:** When you define an account number on TSO, you can specify 1–40 characters. When you define a TSO account number to RACF, you can specify only 1–39 characters.

**NOACCTNUM**

Specifies that you want to delete the user’s default account number. If you delete the default account number from a user’s profile, RACF uses a default value consistent with current TSO defaults when the user logs on to TSO.

**COMMAND | NOCOMMAND**

**COMMAND(command-issued-at-logon)**

Specifies the command to be run during TSO/E logon. TSO/E uses this field to prime the COMMAND field of the logon panel. The command value can contain 1–80 characters and consist of any characters. You can enter the value with or without single quotation marks depending on the following rules:

- If the command value contains parentheses, commas, blanks, or semicolons, enclose the character string in single quotation marks. For example, if the command value is (123), you must enter COMMAND(' (123) ').
- If a single quotation mark is intended to be part of the command value, use two single quotation marks together for each single quotation mark within the string, and enclose the entire string within single quotation marks.

Both uppercase and lowercase characters are accepted and maintained in the case in which they are entered. A user can change the command value, or specify a command if one has not been specified, using the TSO/E logon panel.

**Note:** It is recommended that you use this command for a user who is logged off. If you change the command value for a currently logged-on user ID, the change is overwritten by the TSO/E logoff command processor when the user ID is logged off.

**NOCOMMAND**

Deletes any COMMAND data that was previously saved in the RACF database for this user ID.

**Note:** When you delete this field for a currently logged-on user ID, the field is overwritten by the TSO/E logoff command processor when the user ID is logged off.

**DEST | NODEST**

**DEST(destination-id)**

Specifies the default destination to which the user can route dynamically allocated SYSOUT data sets. The specified value must be 1–7 alphanumeric characters, beginning with an alphabetic or national character.
ALTUSER

NODEST
Specifies that you want to remove any default destination information for this user. Without explicit action by the user to route SYSOUT, the SYSOUT for this user is printed at your system default print location.

HOLDCLASS | NOHOLDCLASS
HOLDCLASS(hold-class)
Specifies the user’s default hold class. The specified value must be 1 alphanumeric character, excluding national characters.

If you specify the TSO operand on the ALTUSER command but do not specify a value for HOLDCLASS, RACF uses a default value consistent with current TSO defaults.

NOHOLDCLASS
Specifies that you want to delete the default hold class from the TSO segment of the user’s profile. If you delete the default hold class from a user’s profile, RACF uses a default value consistent with current TSO defaults when the user logs onto TSO.

JOBCLASS | NOJOBCLASS
JOBCLASS(job-class)
Specifies the user’s default job class. The specified value must be 1 alphanumeric character, excluding national characters.

If you specify the TSO operand on the ALTUSER command but do not specify a value for JOBCLASS, RACF uses a default value consistent with current TSO defaults.

NOJOBCLASS
Specifies that you want to delete the default job class from the TSO segment of the user’s profile. If you delete the default job class from a user’s profile, RACF uses a default value consistent with current TSO defaults when the user logs on to TSO.

MAXSIZE | NOMAXSIZE
MAXSIZE(maximum-region-size)
Specifies the maximum region size that the user can request at logon. The maximum region size is the number of 1024-byte units of virtual storage that TSO can create for the user’s private address space. The specified value must be an integer from 0–65535 if the database is shared with any MVS systems, or 0–2096128 if the database is not shared with any MVS systems.

If you specify the TSO operand on the ALTUSER command but do not specify a value for MAXSIZE, or specify MAXSIZE(0), RACF uses a default value consistent with current TSO defaults.

If values are specified for both MAXSIZE and SIZE and SIZE is greater than MAXSIZE, RACF sets SIZE equal to MAXSIZE. If a value is specified for only SIZE or MAXSIZE and SIZE is greater than MAXSIZE, the operand is ignored.

If your installation is sharing a database between MVS and VM, the administrator is allowed to specify the largest possible value. If your installation is sharing a database between MVS systems, and one of the systems sharing the database is an MVS system, the administrator must not specify a value greater than 65535.
ALTUSER

NOMAXSIZE
Specifies that you want to delete the maximum region size from the TSO segment of the user’s profile. If you delete the maximum region size from a user’s profile, RACF uses a default value consistent with current TSO defaults when the user logs on to TSO.

MSGCLASS | NOMSGCLASS

MSGCLASS(message-class)
Specifies the user’s default message class. The specified value must be one alphanumeric character, excluding national characters.

If you specify the TSO operand on the ALTUSER command but do not specify a value for MSGCLASS, RACF uses a default value consistent with current TSO defaults.

NOMSGCLASS
Specifies that you want to delete the default message class from the TSO segment of the user’s profile. If you delete the default message class from a user’s profile, RACF uses a default value consistent with current TSO defaults when the user logs on to TSO.

PROC | NOPROC

PROC(logon-procedure-name)
Specifies the name of the user’s default logon procedure when logging on through the TSO/E logon panel. The name you specify must be 1–8 alphanumeric characters and begin with an alphabetic character. The name must also be defined as a profile in the TSOPROC general resource class, and the user must be granted READ access to the profile. Otherwise, the user cannot log on to TSO using the specified logon procedure.

A user can change a logon procedure, or specify a logon procedure if one has not been specified, using the TSO/E logon panel. RACF checks the user’s authorization to the specified logon procedure. If the user is authorized to use the logon procedure, RACF stores the name of the procedure in the TSO segment of the user’s profile, and TSO/E uses it as a default value the next time the user logs on to TSO/E. Otherwise, RACF denies the use of the logon procedure.

NOPROC
Specifies that you want to delete the default logon procedure from the TSO segment of the user’s profile. If you delete the default logon procedure from a user’s profile, RACF uses a default value consistent with current TSO defaults when the user logs on to TSO.

SECLABEL | NOSECLABEL

SECLABEL(security-label)
Specifies the user’s security label if the user specifies one on the TSO logon panel.

NOSECLABEL
Specifies that you want to delete the security label from the TSO segment of the user’s profile. If you delete the security label from a user’s TSO segment, RACF uses the security label in the user’s profile the next time the user logs on to TSO.

SIZE | NOSIZE
ALTUSER

SIZE(defaultrgion-size)
Specifies the minimum region size if the user does not request a
region size at logon. The default region size is the number of
1024-byte units of virtual storage available in the user’s private
address space at logon. The specified value must be an integer
from 0–65535 if the database is shared with any MVS systems, or
0–2096128 if the database is not shared with any MVS systems.

A user can change a minimum region size, or specify a minimum
region size if one has not been specified, using the TSO/E logon
panel. RACF stores this value in the TSO segment of the user’s
profile, and TSO/E uses it as a default value the next time the user
logs on to TSO/E.

If values are specified for both MAXSIZE and SIZE and SIZE is
greater than MAXSIZE, RACF sets SIZE equal to MAXSIZE. If a
value is specified for only SIZE or MAXSIZE and SIZE is greater
than MAXSIZE, the operand is ignored.

If your installation is sharing a database between MVS and VM, the
administrator is allowed to specify the largest possible value. If your
installation is sharing a database between MVS systems, and one
of the systems sharing the database is an MVS system, the
administrator must not specify a value greater than 65535.

NOSIZE
Specifies that you want to delete the default minimum region size
from the TSO segment of the user’s profile. If you delete the default
minimum region size from a user’s profile, RACF uses a default
value consistent with current TSO defaults when the user logs on to
TSO.

SYS | NOSYS

SYS(sysout-class)
Specifies the user’s default SYSOUT class. The specified value
must be one alphanumeric character, excluding national characters.

If you specify the TSO operand on the ALTUSER command but do
not specify a value for SYS, RACF uses a default value consistent
with current TSO defaults.

NOSYS
Specifies that you want to delete the default SYSOUT class from
the TSO segment of the user’s profile. If you delete the default
SYSOUT class from a user’s profile, RACF uses a default value
consistent with current TSO defaults when the user logs on to TSO.

UNIT | NOUNIT

UNIT(unit-name)
Specifies the default name of a device or group of devices that a
procedure uses for allocations. The specified value must be 1–8
alphanumeric characters.

NOUNIT
Specifies that you want to delete the default name of a device or
group of devices that a procedure uses for allocations from the TSO
segment of the user’s profile. If you delete this name from a user’s
profile, RACF uses a default value consistent with current TSO
defaults when the user logs on to TSO.
ALTUSER

USERDATA | NOUSERDATA

USERDATA(\textit{user-data})

Specifies optional installation data defined for the user. The specified value must be 4 EBCDIC characters; valid characters are 0–9 and A–F.

\textbf{Note}: When you change this value for a currently logged-on user ID, the change is overwritten by the TSO logoff command processor when the user ID is logged off.

NOUSERDATA

Specifies that you want to delete the installation data previously defined for a user.

NOTSO

Specifies that you are revoking a user's authority to use TSO. RACF deletes TSO logon information from the RACF database for the specified user. However, if the user ID is currently logged on, when the user issues the LOGOFF command the TSO logoff processor restores the TSO segment with default values (except for the USERDATA field which is set to the user's current value). To prevent the TSO segment from being restored, the user ID should be logged off before issuing the ALTUSER NOTSO command.

When you specify NOTSO, the result is the same as if you issue the TSO ACCOUNT command with the DELETE subcommand.

\textbf{UACC(\textit{access-authority})}

Specifies the new default universal access authority for all new resource profiles the user defines while the user's default group or the group specified in the GROUP operand is the user's current connect group. The universal access authorities are ALTER, CONTROL, UPDATE, READ, and NONE. (RACF does not accept EXECUTE access authority with the ALTUSER command.) If you specify UACC without a value, RACF ignores the operand.

This operand is group-related. If a user is subsequently connected to other groups (with the CONNECT command), the user can have a different default universal access authority in each group. Therefore, if the user specifies a different group at logon time or at batch job execution, the user's default UACC is the UACC of the specified group, not the UACC of the user's default group.

UAUDIT | NOUAUDIT

\textbf{UAUDIT}

Specifies that RACF is to log all RACROUTE REQUEST=AUTH and RACROUTE REQUEST=FASTAUTH services that are eligible for logging, and all RACROUTE REQUEST=DEFINE services issued for the user, and all RACF commands (except SEARCH, LISTDSD, LISTGRP, LISTUSER, and RLIST) issued by the user. (When you change a user profile and omit both UAUDIT and NOUAUDIT, the default is NOUAUDIT.)

You must have the AUDITOR attribute, or the user profile must be within the scope of a group in which you have the group-AUDITOR attribute, in order to enter the UAUDIT operand.

If an unauthorized user specifies UAUDIT on the ALTUSER command, none of the operands on the command is processed. RACF issues ICH21005I NOT AUTHORIZED TO SPECIFY UAUDIT, OPERAND IGNORED. The \textbf{System Action} states RACF ignores the operand and continues processing with the
next operand. RACF verifies other operands, but does not process any of them. For more information, see IBM z/OS Security Server RACF Messages and Codes.

NOAUDIT
Specifies that no UAUDIT logging is to be performed. This operand does not override any other auditing options (for example, CMDVIOL specified on SETROPTS) that might be in effect.

You must have the AUDITOR attribute, or the user profile must be within the scope of a group in which you have the group-AUDITOR attribute, to enter the NOAUDIT operand.

WHEN
Specifies the days of the week and the hours in the day when the user is allowed to access the system from a terminal. The day-of-week and time restrictions apply only when a user logs on to the system; that is, RACF does not force the user off the system if the end-time occurs while the user is logged on. Also, the day-of-week and time restrictions do not apply to batch jobs; the user can submit a batch job on any day and at any time.

If you specify the WHEN operand, you can restrict the user’s access to the system to certain days of the week and to a certain time period within each day. For example, you can restrict a user’s access to any one of the following:

- From 9:00 a.m. to 5:00 p.m. (0900:1700). (This would be a daily restriction since days were not also specified.)
- Monday through Friday. (This restriction applies for all 24 hours of Monday, Tuesday, Wednesday, Thursday, and Friday.)
- Monday through Friday from 9:00 a.m. to 5:00 p.m. (0900:1700)

DAYS(day-info)
Specifies days of the week when a user can access the system. The day-info value can be any one of the following:

- ANYDAY
  Specifies that the user can access the system on any day.
- WEEKDAYS
  Specifies that the user can access the system only on weekdays (Monday through Friday).
- day ...
  Specifies that the user can access the system only on the days specified, where day can be MONDAY, TUESDAY, WEDNESDAY, THURSDAY, FRIDAY, SATURDAY, or SUNDAY, and you can specify the days in any order.

Restriction: You cannot specify more than one combination of days and times, even through multiple ALTUSER commands.

Example:
ALTUSER USER127 WHEN(DAYS(MONDAY TUESDAY) TIME(0100:0500))
ALTUSER USER127 WHEN(DAYS(THURSDAY) TIME(0200:0500))

Result:
USER127 is allowed to access the system only on Thursday 2:00–5:00. The preceding DAYS(MONDAY TUESDAY) and TIME(0100:0500) operands are overwritten.

TIME(time-info)
Specifies the time period each day when a user can access the system. The time-info value can be any one of the following:

- ANYTIME
  Specifies that the user can access the system at any time.
Specifies that the user can access the system only during the specified time period. The format of both start-time and end-time is hhmm, where hh is the hour in 24-hour notation (00–23) and mm is the minutes (00–59). Note that 0000 is not a valid time value.

If start-time is greater than end-time, the interval spans midnight and extends into the following day.

If you omit DAYS and specify TIME, the time restriction applies to any day-of-week restriction already indicated in the profile. If you omit TIME and specify DAYS, the day restriction applies to the time restriction already indicated in the profile. If you specify both DAYS and TIME, the user can access the system only during the specified time period and only on the specified days.

If you omit both DAYS and TIME, the time and day restriction remains as it was in the profile.

WORKATTR | NOWORKATTR

WORKATTR

Specifies the user-specific attributes of a unit of work.

z/OS elements or features such as APPC, WLM, and z/OS UNIX might use the WORKATTR segment.

These operands are used by APPC/MVS for SYSOUT created by APPC transactions.

WAACCNT(account-number) | NOWAACNT

Specifies an account number for APPC/MVS processing.

You can specify a maximum of 255 EBCDIC characters. Use the following rules when entering a value for this field:

- If the data contains parentheses, commas, blanks, or semicolons, enclose the character string in single quotation marks. For example, if the data is (123), you must enter WAACCNT(‘123’).
- If a single quotation mark is intended to be part of the data, use two single quotation marks together for each single quotation mark within the string, and enclose the entire string within single quotation marks.

The NOWAACNT suboperand deletes the account number from the user profile.

WAADDRn(address-line-n) | NOWAADDRn

Where n can be 1–4, address-line-n specifies other address lines for SYSOUT delivery. For each line of the address you can specify a maximum of 60 EBCDIC characters. Both uppercase and lowercase characters are accepted and maintained in the case in which they are entered.

Use the following rules when entering a value for this field:

- If the data contains parentheses, commas, blanks, or semicolons, enclose the character string in single quotation marks. For example, if the data is (123), you must enter WAADDR(‘123’).
- If a single quotation mark is intended to be part of the data, use two single quotation marks together for each single quotation mark within the string, and enclose the entire string within single quotation marks.
The NOWAADDR suboperand deletes address line \( n \) from the user profile.

**WABLDG**(building) | NOWABLDG

Specifies the building that SYSOUT information is to be delivered to.

You can specify a maximum of 60 EBCDIC characters. Both uppercase and lowercase characters are accepted and maintained in the case in which they are entered.

Use the following rules when entering a value for this field:

- If the data contains parentheses, commas, blanks, or semicolons, enclose the character string in single quotation marks. For example, if the data is \( (123) \), you must enter `WABLDG(' (123) ')`.
- If a single quotation mark is intended to be part of the data, use two single quotation marks together for each single quotation mark within the string, and enclose the entire string within single quotation marks.

The NOWABLDG suboperand deletes the building from the profile.

**WADEPT**(department) | NOWADEPT

Specifies the department that SYSOUT information is to be delivered to.

You can specify a maximum of 60 EBCDIC characters. Both uppercase and lowercase characters are accepted and maintained in the case in which they are entered.

Use the following rules when entering a value for this field:

- If the data contains parentheses, commas, blanks, or semicolons, enclose the character string in single quotation marks. For example, if the data is \( (123) \), you must enter `WADEPT(' (123) ')`.
- If a single quotation mark is intended to be part of the data, use two single quotation marks together for each single quotation mark within the string, and enclose the entire string within single quotation marks.

The NOWADEPT suboperand deletes the department from the profile.

**WANAME**(name) | NOWANAME

Specifies the name of the user SYSOUT information is to be delivered to.

You can specify a maximum of 60 EBCDIC characters. Both uppercase and lowercase characters are accepted and maintained in the case in which they are entered.

Use the following rules when entering a value for this field:

- If the data contains parentheses, commas, blanks, or semicolons, enclose the character string in single quotation marks. For example, if the data is \( (123) \), you must enter `WANAME(' (123) ')`.
- If a single quotation mark is intended to be part of the data, use two single quotation marks together for each single quotation mark within the string, and enclose the entire string within single quotation marks.

The NOWANAME suboperand deletes the name from the profile.

**WAROOM**(room) | NOWAROOM

Specifies the room SYSOUT information is to be delivered to.

You can specify a maximum of 60 EBCDIC characters. Both uppercase and lowercase characters are accepted and maintained in the case in which they are entered.
ALTUSER

Use the following rules when entering a value for this field:

- If the data contains parentheses, commas, blanks, or semicolons, enclose the character string in single quotation marks. For example, if the data is (123), you must enter WAROOM(' (123 )').

- If a single quotation mark is intended to be part of the data, use two single quotation marks together for each single quotation mark within the string, and enclose the entire string within single quotation marks.

The NOWAROOM suboperand deletes the room from the profile.

NOWORKATTR
Specifies that you want to delete the work attributes previously defined for a user.

Examples

Table 8. ALTUSER examples

| Example 1 | Operation | User IA0 wants to alter the level of group authority from USE to CREATE for user DAF0 in the user's (DAF0's) default group so user DAF0 can define generic profiles for data sets in group RESEARCH. |
| Example 1 | Known | User IA0 is the owner of user DAF0 and has JOIN authority in the group RESEARCH. |
| Example 1 | Command | ALTUSER DAF0 AUTHORITY(CREATE) |
| Example 1 | Defaults | GROUP(RESEARCH) |

Example 2

| Operation | User CD0 wants to correct his name and change his default group to PAYROLL. |
| Known | The default group for user CD0 is RESEARCH. |
| Command | ALTUSER CD0 NAME(CDAVIS) DFLTGRP(PAYROLL) |
| Defaults | None. |

Example 3

| Operation | User IA0 wants to add the FINANCIAL category and the CONFIDENTIAL security level to user ESH25's profile and restrict the user's access to the system to weekdays from 8:00 a.m.–8:00 p.m. |
| Known | User IA0 is connected to group PAYROLL with the group-SPECIAL attribute. Group PAYROLL is user ESH25's default group. |
| Command | ALTUSER ESH25 ADDCATEGORY(FINANCIAL) SECLEVEL(CONFIDENTIAL) WHEN(DAYS(WEEKDAYS)TIME(0800:2000)) |
| Defaults | None. |
ALTUSER

Table 8. ALTUSER examples (continued)

Example 4  
Operation  User RADM02 wants to revoke the user ID of an employee, user D5819, who will be on vacation for three weeks, starting on August 5, 1994. User RADM02 wants to direct the command to run at the local node under the authority of user HICKS and prohibit the command from being automatically directed to other nodes.

Known  Users RADM02 and HICKS have the SPECIAL attribute. Today's date is August 3, 1994. User RADM02 wants to issue the command as a RACF TSO command. Users RADM02 and HICKS have an already established user ID association.

Command  ALTUSER D5819 REVOKE(8/5/94) RESUME(8/26/94) ONLYAT(.HICKS)

Results  The command is only processed on the local node and not automatically directed to any other nodes in the RRSF configuration.

Example 5  
Operation  User RGB01 wants to remove all class authorities and the AUDITOR attribute from USER1, and wants to audit all activity by user USER1.

Known  User RGB01 has the SPECIAL and AUDITOR attributes.

User USER1 is an existing user.

User RGB01 wants to issue the command as a RACF TSO command.

Command  ALTUSER USER1 NOCLAUTH(USER TERMINAL) NOAUDITOR UAUDIT

Defaults  None.

Example 6  
Operation  User RADMIN wants to change the installation-defined information contained in the SJR1 user ID entry, and delete the model name information.

Known  User RADMIN is the owner of user ID SJR1. User RADMIN wants to issue the command as a RACF TSO command.

Command  ALTUSER SJR1 DATA(‘RESOURCE USAGE ADMINISTRATOR NAME TOM P.’) NOMODEL

Defaults  None.

Example 7  
Operation  User VROGERS wants to change default TSO logon information for user BNORTH. User BNORTH requires the following changes:

- A new TSO account number, 12345
- A new TSO logon procedure, LPROC12
- A new SYSOUT data set destination, BL2030
- A new SYSOUT class, Z
- A new maximum region size, 18000.

Known  User VROGERS has the SPECIAL attribute.

- User BNORTH has been defined to RACF with authority to use TSO.
- 12345 has been defined to RACF as a profile in the ACCTNUM general resource class, and user BNORTH has been given READ access to this profile.
- LPROC12 has been defined to RACF as a profile in the TSOPROC general resource class, and user BNORTH has been given READ access to this profile.
- User VROGERS wants to issue the command as a RACF TSO command.

Command  ALTUSER BNORTH TSO(ACCTNUM(12345) PROC(LPROC12) DEST(BL2030) SYS(Z) MAXSIZE(18000))

Defaults  None.
Table 8. ALTUSER examples (continued)

**Example 8**

**Operation**
User MIKEM wants to make the following changes to the profile for user MARTIN:

- Change the default DFP management class to MGMT617
- Change the default DFP storage class to STOR533
- Delete the default DFP data application.

**Known**
- User MIKEM has the SPECIAL attribute.
- User MARTIN has been defined to RACF, and MARTIN’s user profile contains a DFP segment.
- MGMT617 has been defined to RACF as a profile in the MGMTCLAS general resource class, and user MARTIN has been given READ access to this profile.
- STOR533 has been defined to RACF as a profile in the STORCLAS general resource class, and user MARTIN has been given READ access to this profile.
- User MIKEM wants to issue the command as a RACF TSO command.

**Command**
ALTUSER MARTIN DFP(MGMTCLAS(MGMT617) STORCLAS(STOR533) NODATAAPPL))

**Defaults**
None.

**Example 9**

**Operation**
A user with SPECIAL authority wants to make existing z/OS UNIX System Services user CSMITH a superuser and delete PROGRAM from CSMITH’s profile so that the default z/OS UNIX shell program is used when CSMITH enters the TSO/E command OMVS.

**Known**
User CSMITH is already defined to OMVS. The user with SPECIAL authority wants to issue the command as a RACF TSO command.

**Command**
ALTUSER CSMITH OMVS(UID(0) NOPROGRAM)

**Defaults**
None.

**Example 10**

**Operation**
A user with SPECIAL authority wants to make existing z/OS UNIX System Services DCE user, CSMITH, a z/OS UNIX System Services superuser and change the HOMECELL name to /.../hootie.scarol.ibm.com.

**Known**
The DCE UUID for the /.../hootie.scarol.ibm.com cell is 003456ab-ecb7-7de3-ebda-95531ed63dae.

**Command**
ALTUSER CSMITH OMVS(UID(0))
DCE(HOMECELL('/.../hootie.scarol.ibm.com')
HOMEUUID(003456ab-ecb7-7de3-ebda-95531ed63dae))

**Defaults**
None.

**Example 11**

**Operation**
A help desk consultant wants to reset a user’s password.

**Known**
- The consultant is authorized to reset passwords
- The consultant’s RACF user ID (or RACF group associated with the help desk consultant’s user ID) has been permitted by the security administrator with READ access to the RACF FACILITY class profile IRRPASSWORD.RESET.
- The help desk consultant is resetting user JIMBOB’s password.

**Command**
ALTUSER JIMBOB PASSWORD(TEMP012X)

**Defaults**
EXPIRED
Table 8. ALTUSER examples (continued)

Example 12

**Operation**
A help desk consultant wants to reset an application’s password.

**Known**
A help desk consultant has been authorized to reset passwords. The consultant’s RACF user ID (or the RACF group associated with the consultant’s user ID) has been permitted by the security administrator with UPDATE access to the RACF FACILITY class profile IRR.PASSWORD.RESET.

In this example, at the request of operations personnel, the consultant is resetting the user ID associated with an application called CUSTAPP.

The consultant uses the NOEXPIRED operand so the application user ID (CUSTAPP in this example) does not need to change the password when it is logged on.

To reset the application’s password, the consultant enters:

**Command**
```
ALTUSER CUSTAPP PASSWORD(STBR01R) NOEXPIRED
```

**Note:** The password value STBR01R must satisfy the installation’s password quality rules enforced by both SETROPTS and ICHPWX01.

**Defaults**
None.

Example 13

**Operation**
User RACFADM with SPECIAL or UPDATE authority requests the alteration of a RACF user to add Lotus Notes information and to delete the NDS segment from the user’s profile.

**Known**
User RACFADM has SPECIAL authority or UPDATE authority to the desired field within the segment.

**Command**
```
ALTUSER PCUSER2 LNOTES(SNAME(B.B.SMITH)) NONDS
```

**Defaults**
None.

Example 14

**Operation**
User RACFADM with SPECIAL authority adds the user IDs PUBLIC, RACFU00, and USER04. The user ID PUBLIC is then altered and is assigned RESTRICTED access.

**Known**
User RACFADM has SPECIAL authority.

**Command**
```
ADDUSER (PUBLIC RACFU00 USER004) ALTUSER PUBLIC RESTRICTED
ADDS 'RACFU00.*' UACC(READ)
```

**Defaults**
RACFU00, USER004, and PUBLIC have NORESTRICTED access by default.

Example 15

**Operation**
An existing user, whose RACF user profile is RONTOMS, is defining a z/OS Integrated Security Services Network Authentication Service account within the local realm. MAXTKTLFE is not specified, so the value specified on the definition of the local realm KERBDFLT in the REALM class is used.

**Known**
User RONTOMS wants to alter his user profile in order to add z/OS Integrated Security Services Network Authentication Service information.

**Command**
```
ALTUSER RONTOMS KERB(KERBNAME('KerberizedUser'))
PASSWORD(BUNG21R) NOEXPIRED
```

**Defaults**
None.

Example 16

**Operation**
User RACFADMN issues a command to delete the profile that references the EIM domain in the LDAPBIND class for user MRSERVER.

**Known**
The profile in the LDAPBIND class that defines the EIM LDAP values is no longer required for EIM processing.

**Command**
```
ALTUSER MRSERVER EIM(NOLDAPPROF)
```

**Defaults**
None.

Example 17

**Operation**
User RACFADM with SPECIAL authority alters a user’s values for allowable shared and nonshared memory allocation.

**Known**
User RACFADM has SPECIAL authority.

**Command**
```
ALTUSER OMVSUSER OMVS(SHMEMMAX(5M) MEMLIMIT(1G))
```

**Defaults**
None.

**Output**
See [Figure 3](#)
ALTUSER

```
LU OMVSUSER OMVS NORACF
USER=OMVSUSER

OMVS INFORMATION
----------------
UID= 0000000005
CPUTIMEMAX= NONE
ASSIZEMAX= NONE
FILEPROCMA= NONE
PROCUSERMAX= NONE
THREADSMAX= NONE
MMAPAREAMAX= NONE
MEMLIMIT= 1G
SHMEMMAX= 5M
```

Figure 3. Output for ALTUSER Command for OMVS Segment
CONNECT

CONNECT (Connect user to group)

Purpose

Use the CONNECT command to connect a user to a group, modify a user’s connection to a group, or assign the group-related user attributes. If you are creating a connection, defaults are available as stated for each operand. If you are modifying an existing connection, no defaults apply.

RACF date handling: RACF interprets dates with 2-digit years as follows. (The yy value represents the 2-digit year.)
- If 70 < yy <= 99, the date is interpreted as 19yy.
- If 00 <= yy <= 70, the date is interpreted as 20yy.

Issuing options

The following table identifies the eligible options for issuing the CONNECT command:

<table>
<thead>
<tr>
<th>As a RACF TSO command?</th>
<th>As a RACF operator command?</th>
<th>With command direction?</th>
<th>With automatic command direction?</th>
<th>From the RACF parameter library?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

For information on issuing this command as a RACF TSO command, refer to Chapter 3, “RACF TSO commands,” on page 15.

For information on issuing this command as a RACF operator command, refer to Chapter 4, “RACF operator commands,” on page 21.

You must be logged on to the console to issue this command as a RACF operator command.

Related commands

- To list a user’s connections groups, see LISTUSER (List user profile)
- To list the users connected to a group, see LISTGRP (List group profile)
- To remove a user from a group, see REMOVE (Remove user from group)

Authorization required

The specified users and group must already be defined to RACF.

When issuing the CONNECT command as a RACF operator command, you might require sufficient authority to the proper resource in the OPERCMDS class. See z/OS Security Server RACF Security Administrator’s Guide for further information.

To use the CONNECT command, you must have at least one of the following:
- The SPECIAL attribute
- The group-SPECIAL attribute in the group
- The ownership of the group
- JOIN or CONNECT authority in the group.

You cannot give a user a higher level of authority in the group than you have.
To specify the AT keyword, you must have READ authority to the DIRECT.<node> resource in the RRSFDATA class and a user ID association must be established between the specified <node>.<userid> pair(s).

To specify the ONLYAT keyword you must have the SPECIAL attribute, the <userid> specified on the ONLYAT keyword must have the SPECIAL attribute, and a user ID association must be established between the specified <node>.<userid> pair(s) if the user IDs are not identical.

Note: If a user is added to a RACF group as a result of a CONNECT command while the user is logged on, the user must logoff and logon again to use that authority to access resources in classes that have been RACLISTed. In addition, started tasks have to STOP and START to use the new authority. This might include started tasks such as JES2 or JES3.

Syntax

For the key to the symbols used in the command syntax diagrams, see "Syntax of RACF commands and operands" on page 9. The complete syntax of the CONNECT command is:

```
[subsystem-prefix][CONNECT I CO]
  (userid ...)
  [ADSP | NOADSP]
  [AT([node].userid ...) I ONLYAT([node].userid ...)]
  [AUDITOR | NOAUDITOR]
  [AUTHORITY(group-authority)]
  [GROUP(group-name)]
  [GRPACC | NOGRPACC]
  [OPERATIONS | NOOPERATIONS]
  [OWNER(userid or group-name)]
  [RESUME (date)] I NORESUME]
  [REVOKE (date)] I NOREVOKE]
  [SPECIAL | NOSPECIAL]
  [UACC [(access-authority)]]
```

For information on issuing this command as a RACF TSO command, refer to Chapter 3, "RACF TSO commands," on page 15.

For information on issuing this command as a RACF operator command, refer to Chapter 4, "RACF operator commands," on page 21.

Parameters

subsystem-prefix

Specifies that the RACF subsystem is the processing environment of the command. The subsystem prefix can be either the installation-defined prefix for RACF (1–8 characters) or, if no prefix has been defined, the RACF subsystem name followed by a blank. If the command prefix was registered with CPF, you can use the MVS command D OPDATA to display it or you can contact your RACF security administrator.

Only specify the subsystem prefix when issuing this command as a RACF operator command. The subsystem prefix is required when issuing RACF operator commands.
CONNECT

userid
Specifies the RACF-defined user to be connected to, or modified in, the group specified in the GROUP operand. If you are specifying more than one user, you must enclose the user IDs in parentheses.

In general, the maximum number of users you can connect to one group is 5957. See z/OS Security Server RACF Macros and Interfaces for information about how to determine the exact maximum number.

The exception to this is a group that has been defined as a UNIVERSAL group. A UNIVERSAL group may have an unlimited number of users, with USE authority, connected to it for the purpose of resource access.

The number of users in a universal group with authority higher than USE, or with the attributes SPECIAL, OPERATIONS or AUDITOR at the group level, is still limited to 5957.

When displayed with the LISTGRP command, all members of a UNIVERSAL group will be listed. Only users with authority higher than USE or with the attributes SPECIAL, OPERATIONS or AUDITOR at the group level will be shown in the member list.

This operand is required and must be the first operand following CONNECT.

ADSP | NOADSP

ADSP
Specifies that when the user is connected to this group, all permanent tape and DASD data sets created by the user is RACF-protected by discrete profiles.

RACF ignores the ADSP attribute at LOGON/job initiation if SETROPTS NOADSP is in effect.

NOADSP
Specifies that the user is not to have the ADSP attribute. If you are creating a connection and omit both ADSP and NOADSP, NOADSP is the default. A user attribute of ADSP specified on the ADDUSER or ALTUSER command overrides NOADSP as a connect attribute.

AT | ONLYAT
The AT and ONLYAT keywords are only valid when the command is issued as a RACF TSO command.

AT([node].userid ...)
Specifies that the command is to be directed to the node specified by node, where it runs under the authority of the user specified by userid in the RACF subsystem address space.

If node is not specified, the command is directed to the local node.

ONLYAT([node].userid ...)
Specifies that the command is to be directed only to the node specified by node where it runs under the authority of the user specified by userid in the RACF subsystem address space.

If node is not specified, the command is directed only to the local node.

AUDITOR | NOAUDITOR

AUDITOR
Specifies that the user is to have the group-AUDITOR attribute when connected to this group.
To enter the AUDITOR operand, you must have either the SPECIAL attribute or the group-SPECIAL attribute in the group to which you are connecting or modifying the user’s profile.

NOAUDITOR
Specifies that the user is not to have the group-AUDITOR attribute when connected to this group. When you are creating a connection and omit both AUDITOR and NOAUDITOR, NOAUDITOR is the default. If you are modifying an existing connection, you must have either the SPECIAL attribute or the group-SPECIAL attribute in the group in which you are modifying the user’s profile.

A user attribute of AUDITOR specified on the ADDUSER or ALTUSER command overrides NOAUDITOR as a connect attribute.

AUTHORITY(

) Specifies the level of authority the user is to have in the group. The valid group authority values are USE, CREATE, CONNECT, and JOIN, as described in "Group authorities" on page 13. If you are creating a connection and omit AUTHORITY or enter it without a value, the default is USE.

You cannot give a user a higher level of authority in the group than you have.

GROUP(

) Specifies a RACF-defined group. If you omit this operand, the user is connected to or modified in your current connect group.

Note: RACF allows you to connect a user to more than 300 groups, which is the same as NGROUPS_MAX variable defined in the POSIX standard, but when z/OS UNIX group information is requested, only up to the first 300 z/OS UNIX groups that have GIDs are associated with the process or user.

The first 300 z/OS UNIX groups that have GIDs to which a user is connected are used by z/OS UNIX. LISTUSER displays the groups in the order that RACF examines them when determining which of the user’s groups are z/OS UNIX groups.

In addition, the number of users connected to a group should be within the limits allowed by the NFS client for remote access to files. See z/OS UNIX System Services Planning for information on NGROUPS_MAX.

GRPACC | NOGRPACC

GRPACC
Specifies that when the user is connected to this group, any group data sets defined by the user are automatically accessible to other users in the group. The group whose name is used as the high-level qualifier of the data set name (or the qualifier supplied by a command installation exit) has UPDATE access authority to the data set.

NOGRPACC
Specifies that the user is not to have the GRPACC attribute. If you are creating a connection and omit both GRPACC and NOGRPACC, NOGRPACC is the default. A user attribute of GRPACC specified on the ADDUSER or ALTUSER command overrides NOGRPACC as a connect attribute.

OPERATIONS | NOOPERATIONS

OPERATIONS
Specifies that the user is to have the group-OPERATIONS attribute when
CONNECT

connected to this group. The group-OPERATIONS user has authorization to
do maintenance operations on all RACF-protected DASD data sets, tape
volumes, and DASD volumes within the scope of the group unless the
access list for a resource specifically limits the OPERATIONS user to an
access authority that is less than the operation requires.

You establish the lower access authority for the group-OPERATIONS user
through the PERMIT command.

To enter the OPERATIONS operand, you must have the SPECIAL attribute
or the group-SPECIAL attribute in the group to which you are connecting or
modifying the user's profile.

NOOPERATIONS
Specifies that the user is not to have the group-OPERATIONS attribute in
this group. If you are creating a connection and omit both OPERATIONS
and NOOPERATIONS, NOOPERATIONS is the default. If you are modifying
an existing connection, you must have the SPECIAL attribute or the
group-SPECIAL attribute in the group in which you are modifying the user's
profile.

A user attribute of OPERATIONS specified on the ADDUSER or ALTUSER
command overrides NOOPERATIONS as a connect attribute.

OWNER(userid or group-name)
Specifies a RACF-defined user or group to be assigned as the owner of the
connect profile. If you are creating a connection and you do not specify an
owner, you are defined as the owner of the connect profile.

RESUME | NORESUME

RESUME[(date)]
Specifies that the user, when connected to the group specified on the
GROUP operand, is to be allowed to access the system again. You
normally use RESUME to restore access to the system that has been
prevented by a prior REVOKE operand. (RESUME, using the current date,
is also the default when you are using the CONNECT command to create
an initial connection between a user and this group.)

If you specify a date, RACF does not allow the user to access the system
until the date you specify. The date must be a future date; if it is not, you
are prompted to provide a future date.

Between the time you specify the RESUME and the time the RESUME
takes effect, the RESUME is called a pending resumption (or a pending
RESUME).

You specify a date in the form mm/dd/yy, and you need not specify leading
zeros; specifying 9/1/06 is the same as specifying 09/01/06. The date must
be a future date; if it is not, you are prompted to provide a future date.
RACF interprets dates as 20yy when yy is less than 71, and 19yy when yy
is 71 or higher. So, 09/01/94 would be in the year 1994, and 09/01/14
would be in the year 2014.

If you specify RESUME without a date, the RESUME takes effect
immediately.

When no REVOKE is in effect for the user, RACF ignores the RESUME
operand and issues a message.
Notes:

1. If you use the ALTUSER command to issue a REVOKE for a user, you must use the ALTUSER command to issue the corresponding RESUME. Issuing RESUME on the CONNECT command does not restore access revoked on the ALTUSER command.

2. If you specify both REVOKE(date) and RESUME(date), RACF acts on them in date order. For example, if you specify RESUME(8/19/06) and REVOKE(8/5/06), RACF prevents the user from accessing the system from August 5, 2006, to August 18, 2006. On August 19, the user can again access the system.

   If a user is already revoked and you specify RESUME(8/5/06) and REVOKE(8/19/06), RACF allows the user to access the system from August 5, 2006, to August 18, 2006. On August 19, RACF prevents the user from accessing the system.

3. If RACF detects a conflict between REVOKE and RESUME (for example, you specify both without a date), RACF uses REVOKE.

4. To clear the RESUME date field, specify NORESUME.

5. To successfully resume a user whose revoke date has passed, you must specify NOREVOKE to clear the revoke date as well as specifying the RESUME keyword.

6. Downlevel systems sharing the RACF database should not be affected by the changes to REVOKE and RESUME processing. A user who is considered revoked on a z/OS V1R7 system should also be considered revoked on a downlevel system.

NORESUME

Specifies that RACF is to clear the RESUME date field in the user’s group connection. You can use the NORESUME option to cancel the pending resumption (of a user’s group connection) that resulted from a previous CONNECT command specified with RESUME(date).

REVOKE | NOREVOKE

REVOKE([date])

Specifies that RACF is to prevent the user from accessing the system by attempting to connect to the group specified on the GROUP operand. The user’s profile and data sets are not deleted from the RACF database.

If you specify a date, RACF does not prevent the user from accessing the system until the date you specify. The date must be a future date; if it is not, you are prompted to provide a future date.

You specify a date in the form mm/dd/yy, and you need not specify leading zeros; specifying 9/1/06 is the same as specifying 09/01/06. The date must be a future date; if it is not, you are prompted to provide a future date.

RACF interprets dates as 20yy when yy is less than 71, and 19yy when yy is 71 or higher. So, 09/01/94 would be in the year 1994, and 09/01/14 would be in the year 2014.

Between the time you specify the REVOKE and the time the REVOKE takes effect, the REVOKE is called a pending revocation (or a pending REVOKE).

When you specify REVOKE without a date, the following conditions apply:

- The REVOKE takes effect the next time the user tries to log onto the system.
CONNECT

- Any pending RESUME date remains in effect unless you also specify NORESUME.

Important: To permanently revoke system access, specify both REVOKE and NORESUME.

When a REVOKE is already in effect for the user, RACF ignores the REVOKE operand and issues a message.

Notes:
1. If you specify both REVOKE(date) and RESUME(date), RACF acts on them in date order. For example, if you specify RESUME(8/19/06) and REVOKE(8/5/06), RACF prevents the user from accessing the system from August 5, 2006, to August 18, 2006. On August 19, the user can again access the system.

   If a user is already revoked and you specify RESUME(8/5/06) and REVOKE(8/19/06), RACF allows the user to access the system from August 5, 2006, to August 18, 2006. On August 19, RACF prevents the user from accessing the system.

2. If RACF detects a conflict between REVOKE and RESUME (for example, you specify both without a date), RACF uses REVOKE.

3. To clear the REVOKE date field, specify NOREVOKE.

4. Downlevel systems sharing the RACF database should not be affected by the changes to REVOKE and RESUME processing. A user who is considered revoked on a z/OS V1R7 system should also be considered revoked on a downlevel system.

NOREVOKE

Specifies that RACF is to clear the REVOKE date field in the user’s group connection. You can use the NOREVOKE option to cancel the pending revocation (of a user's group connection) that resulted from a previous CONNECT command specified with REVOKE(date).

To successfully resume a user whose revoke date has passed, you must specify NOREVOKE to clear the revoke date as well as specifying the RESUME keyword.

The NOREVOKE option does not resume the user’s group connection after it was revoked by the CONNECT REVOKE command.

SPECIAL | NOSPECIAL

SPECIAL

Specifies that the user is to have the group-SPECIAL attribute when connected to this group. To enter the SPECIAL operand, you must have the SPECIAL attribute or the group-SPECIAL attribute in the group to which you are connecting or modifying the user’s profile.

NOSPECIAL

Specifies that the user is not to have the group-SPECIAL attribute. If you are creating a connection and omit both SPECIAL and NOSPECIAL, NOSPECIAL is the default. If you are modifying an existing connection, you must have the SPECIAL attribute or the group-SPECIAL attribute in the group in which you are modifying the user’s profile.

A user attribute of SPECIAL specified on the ADDUSER or ALTUSER command overrides NOSPECIAL as a connect attribute.

UACC([access-authority])

Specifies the default value for the universal access authority for all new
resource profiles the user defines while the specified group is the user's current
connect group. The universal access authorities are ALTER, CONTROL,
UPDATE, READ, and NONE. (RACF does not accept EXECUTE access
authority with the CONNECT command.) If you are creating a connection and
omit UACC or enter it without a value, the default is NONE.

This operand is group-related. The user can have a different default universal
access authority in each of the groups to which the user is connected (with the
CONNECT command).

Examples

Table 9. CONNECT examples

Example 1

Operation  User WJE10 wants to connect users AFG5 and GMD2 to group PAYROLL and
to make PAYROLL the owner of the connect profiles.

Known  User WJE10 has JOIN authority to group PAYROLL.

User WJE10 is currently connected to group PAYROLL.

Users AFG5 and GMD2 are defined to RACF but not connected to group
PAYROLL.

User WJE10 wants to issue the command as a RACF TSO command.

Command  CONNECT (AFG5 GMD2) OWNER(PAYROLL)

Defaults  GROUP(PAYROLL) AUTHORITY(USE) UACC(NONE) NOADSP NOGRPACC

Example 2

Operation  User WRH0 wants to CONNECT user PDJ6 to group RESEARCH with
CREATE authority and universal access of UPDATE. User WRH0 wants to
direct the command to run under the authority of user EMWIN at node RALNC.

Known  User EMWIN at RALNC has CONNECT authority to group RESEARCH.

RESEARCH is not the default group of user EMWIN at RALNC.

User PDJ6 is defined to RACF on node RALNC but is not connected to group
RESEARCH.

User WRH0 wants to issue the command as a RACF TSO command.

Command  CONNECT PDJ6 GROUP(RESEARCH) AUTHORITY(CREATE) UACC(UPDATE)
AT(RALNC.EMWIN)

Defaults  NOGRPACC RESUME NOOPERATIONS NOSPECIAL NOAUDITOR NOADSP
OWNER(WRH0)

Example 3

Operation  User IRB01 wants to revoke the user ID of an employee, user D5819, who will
be on vacation for three weeks, starting on August 5, 1994.

Known  User IRB01 is the owner of the profile for user D5819. Today's date is August
3, 1994. User IRB01 wants to issue the command as a RACF operator
command, and the RACF subsystem prefix is @.

Command  @CONNECT D5819 REVOKE(8/5/94) RESUME(8/26/94)

Defaults  None.
DELDSD

DELDSD (Delete data set profile)

Purpose

Use the DELDSD command to remove RACF protection from tape or DASD data sets that are protected by either discrete or generic profiles.

When RACF-protection is removed from a data set protected by a discrete profile:
- The RACF indicator for the data set is turned off. For a DASD data set, the indicator is in the DSCB for a non-VSAM data set or in the catalog entry for a VSAM data set. For a tape data set, the indicator is in the TVTOC entry for the data set in the corresponding TAPEVOL profile.
- The data set profile is deleted from the RACF database. (Note that the data set itself is not physically deleted or scratched.)
- If all the data sets in the TVTOC have expired, then RACF deletes the TAPEVOL profiles and the associated tape DATASET profiles.

To remove RACF protection from a non-VSAM DASD data set that is protected by a discrete profile, the data set must be online and not currently in use. For a VSAM data set that is protected by a discrete profile, the catalog for the data set must be online. The VSAM data set itself must also be online if the VSAM catalog recovery option is being used. If the required data set or catalog is not online, the DELDSD command processor requests that the volume be mounted if you have the TSO MOUNT authority.

Changes made to discrete profiles take effect after the DELDSD command is processed. Changes made to generic profiles do not take effect until one or more of the following steps is taken:
- The user of the data set issues the LISTDSD command:
  LISTDSD DA(data-set-protected-by-the-profile) GENERIC
  
  **Note:** Use the data set name, not the profile name.
- The security administrator issues the SETROPTS command:
  SETROPTS GENERIC(DATASET) REFRESH
  See SETROPTS command for authorization requirements.
- The user of the data set logs off and logs on again.

  **Note:** For more information, refer to [z/OS Security Server RACF Security Administrator’s Guide](#).

Issuing options

The following table identifies the eligible options for issuing the DELDSD command:

<table>
<thead>
<tr>
<th>As a RACF TSO command?</th>
<th>As a RACF operator command?</th>
<th>With command direction?</th>
<th>With automatic command direction?</th>
<th>From the RACF parameter library?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

For information on issuing this command as a RACF TSO command, refer to Chapter 3, “RACF TSO commands,” on page 15.
DELDSD

For information on issuing this command as a RACF operator command, refer to Chapter 4, “RACF operator commands,” on page 21.

You must be logged on to the console to issue this command as a RACF operator command.

Related commands

- To create a data set profile, see ADDSD (Add data set profile).
- To change a data set profile, see ALTDSD (Alter data set profile).
- To display a data set profile, see LISTDSD (List data set profile).
- To obtain a list of data set profiles, see SEARCH (Search RACF database).

Authorization required

When issuing the DELDSD command as a RACF operator command, you might require sufficient authority to the proper resource in the OPERCMDS class. See z/OS Security Server RACF Security Administrator’s Guide for further information.

To remove RACF protection from a data set or to delete a generic data set profile, you must have sufficient authority over the data set. RACF performs authorization checking in the following sequence until you meet one of these conditions:

- You have the SPECIAL attribute.
- The data set profile is within the scope of a group in which you have the group-SPECIAL attribute.
- The high-level qualifier of the profile name (or the qualifier supplied by a command installation exit) is your user ID.
- You are the owner of the profile.
- For a discrete profile, you are on the access list with ALTER authority.
- For a discrete profile, your group or one of your groups (if checking list of groups is active) is on the access list and has ALTER authority.
- For a discrete profile, the universal access authority is ALTER.

To specify the AT keyword, you must have READ authority to the DIRECT.node resource in the RRSFDATA class and a user ID association must be established between the specified node.userid pair(s).

To specify the ONLYAT keyword you must have the SPECIAL attribute, the userid specified on the ONLYAT keyword must have the SPECIAL attribute, and a user ID association must be established between the specified node.userid pair(s) if the user IDs are not identical.

Syntax

For the key to the symbols used in the command syntax diagrams, see “Syntax of RACF commands and operands” on page 9. The complete syntax of the DELDSD command is:
DELDSD

[subsystem-prefix][DELDSD | DD]

(profile-name...)

[ AT([node].userid ...) | ONLYAT([node].userid ...) ]
[ GENERIC | NOSET | SET ]
[ VOLUME(volume-serial) ]

Note: If you specify a profile name containing generic characters, RACF ignores the VOLUME, SET and NOSET operands.

For information on issuing this command as a RACF TSO command, refer to Chapter 3, “RACF TSO commands,” on page 15.

For information on issuing this command as a RACF operator command, refer to Chapter 4, “RACF operator commands,” on page 21.

Parameters

subsystem-prefix
Specifies that the RACF subsystem is the processing environment of the command. The subsystem prefix can be either the installation-defined prefix for RACF (1–8 characters) or, if no prefix has been defined, the RACF subsystem name followed by a blank. If the command prefix was registered with CPF, you can use the MVS command D OPDATA to display it or you can contact your RACF security administrator.

Only specify the subsystem prefix when issuing this command as a RACF operator command. The subsystem prefix is required when issuing RACF operator commands.

profile-name ...
Specifies the name of the discrete or generic profile. If you specify more than one profile, the list must be enclosed in parentheses.

This operand is required and must be the first operand following DELDSD.

Note: Because RACF uses the RACF database and not the system catalog, you cannot use alias data set names.

AT \ ONLYAT
The AT and ONLYAT keywords are only valid when the command is issued as a RACF TSO command.

AT([node].userid ...)
Specifies that the command is to be directed to the node specified by node, where it runs under the authority of the user specified by userid in the RACF subsystem address space.

If node is not specified, the command is directed to the local node.

ONLYAT([node].userid ...)
Specifies that the command is to be directed only to the node specified by node where it runs under the authority of the user specified by userid in the RACF subsystem address space.

If node is not specified, the command is directed only to the local node.

GENERIC | NOSET | SET
If you do not specify GENERIC, NOSET, or SET, the default value is SET.
DELDSD

**GENERIC**
Specifies that RACF is to treat the profile name as a generic name, even if it does not contain any generic characters.

**NOSET | SET**
Specifies whether the RACF indicator should be set off or not.

If the profile name contains a generic character or if you specify GENERIC, RACF ignores this operand.

**NOSET**
Specifies that RACF is not to turn off the RACF indicator for the data set.

Use NOSET when you are transferring a RACF-indicated data set to another system where it is also to be RACF-protected. Leaving the indicator on prevents unauthorized access to the data set until it can be redefined on the new system. (To delete multiple data set profiles, see Example 2 for the SEARCH command.)

When you specify NOSET for a tape data set protected by a discrete profile, RACF deletes the discrete profile but retains the TVTOC entry for the data set name. You can then use a generic profile to protect the data set.

If you specify NOSET, the volumes on which the data set or catalog resides need not be online.

To use NOSET, you must have the SPECIAL attribute, the data set profile must be within the scope of a group in which you have the group-SPECIAL attribute, or the high-level qualifier of the data set name (or the qualifier supplied by the naming conventions table or by a command installation exit) must be your user ID.

**SET**
Specifies that RACF is to turn off the RACF indicator for the data set. Use SET, which is the default value, when you are removing RACF protection for a data set. If the indicator is already off, the command fails.

**VOLUME**(volume-serial)
Specifies the volume on which the tape data set, the non-VSAM DASD data set, or the catalog for the VSAM data set resides.

If you specify this operand and volume-serial does not appear in the profile for the data set, the command fails.

If the data set name appears more than once in the RACF database and you do not specify VOLUME, the command fails. If the data set name appears only once and you do not specify VOLUME, no volume serial number checking is performed, and processing continues.

If the profile name contains a generic character or if you specify GENERIC, RACF ignores this operand.
DELDSD

Examples

Table 10. DELDSD examples

<table>
<thead>
<tr>
<th>Example</th>
<th>Operation</th>
<th>Known</th>
<th>Command</th>
<th>Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>User EH0 wants to remove discrete profile RACF protection from data set CD0.DEPT1.DATA. User EH0 wants to direct the command to run at node CPPD0 under the authority of user GCP02 and prohibit the command from being automatically directed to other nodes.</td>
<td>User GCP02 at CPPD0 owns data set CD0.DEPT1.DATA. User EH0 wants to issue the command as a RACF TSO command. Users EH0 and GCP02 at CPPD0 have an already established user ID association. Users EH0 and GCP02 at CPPD0 have the SPECIAL attribute.</td>
<td>DELDSD 'CD0.DEPT1.DATA' ONLYAT(CPPD0,GCP02)</td>
<td>The command is only processed at node CPPD0 and not automatically directed to any other nodes in the RRSF configuration.</td>
</tr>
<tr>
<td>2</td>
<td>User KLE05 wants to enter a RACF TSO command to remove discrete profile protection from data set KLE05.DUPDS1.DATA. The data set is a duplicate data set, and the user wants to remove the profile for the data set on volume DU2 without turning off the RACF indicator.</td>
<td></td>
<td>DELDSD DUPDS1.DATA VOLUME(DU2) NOSET</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>User JTB01 wants to delete the generic profile and remove RACF protection from the data set or sets protected by the profile SALES.*.DATA</td>
<td>User JTB01 has the group-SPECIAL attribute in group SALES. User JTB01 wants to issue the command as a RACF operator command, and the RACF subsystem prefix is @.</td>
<td>@DELDSD 'SALES.*.DATA'</td>
<td></td>
</tr>
</tbody>
</table>

Defaults

None.
DELGROUP (Delete group profile)

Purpose

Use the DELGROUP command to delete a group and its relationship to its superior group from RACF.

There are, however, other places in the RACF database where the group name might appear, and DELGROUP processing does not delete these other occurrences of the group name. For example, the group name could be in the access list for any resource. You can use the RACF Remove ID utility (IRRRID00) to remove all occurrences of a group name.

The DELGROUP command does not work for a UNIVERSAL group, in most cases. To delete a UNIVERSAL group, the RACF Remove ID Utility (IRRRID00) should be used.

For information on using the RACF remove ID utility, see z/OS Security Server RACF Security Administrator’s Guide

Issuing options

The following table identifies the eligible options for issuing the DELGROUP command:

<table>
<thead>
<tr>
<th>As a RACF TSO command?</th>
<th>As a RACF operator command?</th>
<th>With command direction?</th>
<th>With automatic command direction?</th>
<th>From the RACF parameter library?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

For information on issuing this command as a RACF TSO command, refer to Chapter 3, “RACF TSO commands,” on page 15.

For information on issuing this command as a RACF operator command, refer to Chapter 4, “RACF operator commands,” on page 21.

You must be logged on to the console to issue this command as a RACF operator command.

Related commands

- To add a group profile to the RACF database, see ADDGROUP (Add group profile)
- To change a group profile in the RACF database, see ALTGROUP (Alter group profile)
- To connect a user to a group, see CONNECT (Connect user to group)
- To list information related to a group profile, see LISTGRP (List group profile)
- To remove a user from a group profile, see REMOVE (Remove user from group)
- To obtain a list of group profiles, see SEARCH (Search RACF database)

Authorization required

When issuing the DELGROUP command as a RACF operator command, you might require sufficient authority to the proper resource in the OPERCMDS class. See z/OS Security Server RACF Security Administrator’s Guide for further information.
To use the DELGROUP command, at least one of the following must be true:

- You must have the SPECIAL attribute
- The group to be deleted must be within the scope of a group in which you have the group-SPECIAL attribute
- You must be the owner of the superior group
- You must have JOIN authority in the superior group
- You must be the owner of the group to be deleted

To specify the AT keyword, you must have READ authority to the DIRECT.node resource in the RRSFDATA class and a user ID association must be established between the specified node.userid pair(s).

To specify the ONLYAT keyword you must have the SPECIAL attribute, the userid specified on the ONLYAT keyword must have the SPECIAL attribute, and a user ID association must be established between the specified node.userid pair(s) if the user IDs are not identical.

**Syntax**

For the key to the symbols used in the command syntax diagrams, see "Syntax of RACF commands and operands" on page 9. The complete syntax of the DELGROUP command is:

```plaintext
[subsystem-prefix][DELGROUP | DG]
  (group-name ...)
  [ AT([node].userid ...) | ONLYAT([node].userid ...) ]
```

For information on issuing this command as a RACF TSO command, refer to Chapter 3, “RACF TSO commands,” on page 15.

For information on issuing this command as a RACF operator command, refer to Chapter 4, “RACF operator commands,” on page 21.

**Parameters**

- **subsystem-prefix**
  - Specifies that the RACF subsystem is the processing environment of the command. The subsystem prefix can be either the installation-defined prefix for RACF (1–8 characters) or, if no prefix has been defined, the RACF subsystem name followed by a blank. If the command prefix was registered with CPF, you can use the MVS command D OPDATA to display it or you can contact your RACF security administrator.
  - Only specify the subsystem prefix when issuing this command as a RACF operator command. The subsystem prefix is required when issuing RACF operator commands.

- **group-name**
  - Specifies the name of the group whose profile is to be removed from the RACF database. If you are deleting more than one group, you must enclose the list of group names in parentheses.
  - You must enter at least one group name. For each group name you enter, the following conditions must exist:
    - The group must be defined to RACF.
• The group must not have any subgroups.
• The group must not have any group data sets (data sets whose names are qualified by the group name or begin with the value supplied by an installation exit).
• The group must not have any users connected to it.

**AT | ONLYAT**
The AT and ONLYAT keywords are only valid when the command is issued as a RACF TSO command.

**AT**
AT([node].userid ...)
Specifies that the command is to be directed to the node specified by node, where it runs under the authority of the user specified by userid in the RACF subsystem address space.
If node is not specified, the command is directed to the local node.

**ONLYAT**
ONLYAT([node].userid ...)
Specifies that the command is to be directed only to the node specified by node where it runs under the authority of the user specified by userid in the RACF subsystem address space.
If node is not specified, the command is directed only to the local node.

**Examples**

*Table 11. DELGROUP example*

<table>
<thead>
<tr>
<th>Example</th>
<th>Operation</th>
<th>Known</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>User WJE10 wants to delete subgroups DEPT1 and DEPT2 from group PAYROLL.</td>
<td>User WJE10 has JOIN authority to group PAYROLL.</td>
<td>DELGROUP (DEPT1 DEPT2)</td>
</tr>
<tr>
<td></td>
<td>DEPT1 and DEPT2 are subgroups of group PAYROLL.</td>
<td>Neither DEPT1 nor DEPT2 have any subgroups or users connected to them.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>In addition, neither group has any group data sets.</td>
<td>User WJE10 wants to issue the command as a RACF TSO command.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Defaults None.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
DELUSER

DELUSER (Delete user profile)

Purpose

Use the DELUSER command to delete a user from RACF.

This command removes the user’s profile and all user-to-group connections for the user. (The connect profiles define the user’s connections to various RACF groups.)

There are, however, other places in the RACF database where the user’s user ID might appear, and the DELUSER command does not delete the user ID from all these places. Specifically, the user could be the owner of a group, the owner of a user’s profile, the owner of a group data set, or in an access list for any resource. Before issuing DELUSER, you must first issue the REMOVE command to assign new owners for any group data sets the user owns in groups other than his default group. You can use the RACF remove ID utility (IRRRID00) to remove all of the occurrences of a user ID. For information on using the RACF remove ID utility, see z/OS Security Server RACF Security Administrator's Guide.

You can use the DELUSER command to delete a TSO user from the RACF database. However, you have no way of knowing if the TSO user is logged on to TSO at the time you issue the DELUSER command. As a result, if the user is logged on to TSO, the user remains active until logging off. Therefore, you might consider having the console operator examine any logons (or jobs) that are active for the TSO user and cancel those that should not be allowed to continue.

The DELUSER command supports digital certificates. If the command issuer is authorized to delete the user profile, and the DELUSER command processor has decided that the user profile can be deleted, the profiles in the DIGTCERT, DIGTRING, or DIGTNMAP classes that describe certificates, private key information, key rings, or certificate mappings associated with the user profile are also deleted. When determining what certificates to delete, the list of certificates from the user profile is used. Certificates that are to be deleted as a result of DELUSER processing are removed from any rings they are connected to at the time the DELUSER command was issued. Likewise, rings that are to be deleted as a result of DELUSER processing have all certificates connected to them removed prior to being deleted. No additional authority checking is done. Authority to the IRR.DIGTCERT.function resource is not required. If an error is encountered by DELUSER while attempting to delete a DIGTCERT, DIGTRING, or DIGTNMAP profile, the DELUSER command is terminated without attempting to delete the user profile. If the error indicates that the template is downlevel, an error message is issued and the user profile is deleted.

Restrictions:

- User IDs with mixed-case characters, such as irrcerta, irrsitec, and irrmulti which are associated with digital certificates, cannot be specified as userid in the DELUSER command because DELUSER cannot process mixed-case user IDs.

- Do not issue a DELUSER command for user ID that has a distributed identity filter (contained in an IDIDMAP profile) associated with it. The command will fail with error message ICH04018I. You must first delete the distributed identity filter. To do this, issue the RACMAP LISTMAP command for the user ID to examine the name filter and determine its label name, and then issue the RACMAP DELMAP command.
Issuing options

The following table identifies the eligible options for issuing the DELUSER command:

<table>
<thead>
<tr>
<th></th>
<th>As a RACF TSO command?</th>
<th>As a RACF operator command?</th>
<th>With command direction?</th>
<th>With automatic command direction?</th>
<th>From the RACF parameter library?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

For information on issuing this command as a RACF TSO command, refer to Chapter 3, “RACF TSO commands,” on page 15.

For information on issuing this command as a RACF operator command, refer to Chapter 4, “RACF operator commands,” on page 21.

You must be logged on to the console to issue this command as a RACF operator command.

Related commands

- To add a user profile to the RACF database, see ADDUSER (Add user profile).
- To change a user profile in the RACF database, see ALTUSER (Alter user profile).
- To list information in a user profile, see LISTUSER (List user profile).
- To administer user ID associations, see RACLINK (Administer user ID associations).

Authorization required

When issuing the DELUSER command as a RACF operator command, you might require sufficient authority to the proper resource in the OPERCMDS class. See z/OS Security Server RACF Security Administrator's Guide for further information.

To use the DELUSER command, at least one of the following must be true:

- You must have the SPECIAL attribute.
- The user profile to be deleted must be within the scope of a group in which you have the group-SPECIAL attribute.
- You must be the owner of the user’s profile.

Note: JOIN authority in the user’s default group is not sufficient authority to delete the user from RACF.

To specify the AT keyword, you must have READ authority to the DIRECT.node resource in the RRSFDATA class and a user ID association must be established between the specified node.userid pair(s).

To specify the ONLYAT keyword you must have the SPECIAL attribute, the userid specified on the ONLYAT keyword must have the SPECIAL attribute, and a user ID association must be established between the specified node.userid pair(s) if the user IDs are not identical.

Syntax

For the key to the symbols used in the command syntax diagrams, see “Syntax of RACF commands and operands” on page 9. The complete syntax of the DELUSER
DELUSER

command is:

```
[subsystem-prefix][DELUSER | DU]
(userid ...)
[ AT([node].userid ...) | ONLYAT([node].userid ...) ]
```

For information on issuing this command as a RACF TSO command, refer to Chapter 3, “RACF TSO commands,” on page 15.

For information on issuing this command as a RACF operator command, refer to Chapter 4, “RACF operator commands,” on page 21.

Parameters

`subsystem-prefix`

Specifies that the RACF subsystem is the processing environment of the command. The subsystem prefix can be either the installation-defined prefix for RACF (1–8 characters) or, if no prefix has been defined, the RACF subsystem name followed by a blank. If the command prefix was registered with CPF, you can use the MVS command D OPDATA to display it or you can contact your RACF security administrator.

Only specify the subsystem prefix when issuing this command as a RACF operator command. The subsystem prefix is required when issuing RACF operator commands.

`userid`

Specifies the user ID of the user whose profile is to be deleted from the RACF database. If you are deleting more than one user, you must enclose the list of user IDs in parentheses. You must enter at least one user ID. For each user ID you enter, the following conditions must exist:
- The user must be defined to RACF.
- The user must not have any user data sets defined to RACF. (User data sets are data sets whose names are qualified by the user ID of the user being deleted or begin with the value supplied by an installation exit.)
- The user cannot have any user ID associations defined. User ID associations for a user must be deleted before the user can be deleted.

`AT` | `ONLYAT`

The AT and ONLYAT keywords are only valid when the command is issued as a RACF TSO command.

`AT([node].userid ...)`

Specifies that the command is to be directed to the node specified by `node`, where it runs under the authority of the user specified by `userid` in the RACF subsystem address space.

If `node` is not specified, the command is directed to the local node.

`ONLYAT([node].userid ...)`

Specifies that the command is to be directed only to the node specified by `node` where it runs under the authority of the user specified by `userid` in the RACF subsystem address space.

If `node` is not specified, the command is directed only to the local node.
Examples

Table 12. DELUSER examples

Example 1

<table>
<thead>
<tr>
<th>Operation</th>
<th>User WJE10 wants to delete user AEH0 from RACF.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Known</td>
<td>User AEH0 is defined to RACF.</td>
</tr>
<tr>
<td></td>
<td>User AEH0 is not the owner of any RACF profiles.</td>
</tr>
<tr>
<td></td>
<td>User WJE10 is connected to group PAYROLL (and is the owner of user AEH0) with the group-SPECIAL attribute.</td>
</tr>
<tr>
<td></td>
<td>User WJE10 wants to issue the command as a RACF TSO command.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Command</th>
<th>DELUSER AEH0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Defaults</td>
<td>None.</td>
</tr>
</tbody>
</table>

Example 2

<table>
<thead>
<tr>
<th>Operation</th>
<th>User SPB1 wants to delete user CA00 from RACF.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Known</td>
<td>User CA00 is defined to RACF.</td>
</tr>
<tr>
<td></td>
<td>User SPB1 is not the owner of any RACF profiles.</td>
</tr>
<tr>
<td></td>
<td>User SPB1 is connected to group PAYROLL (and is the owner of user CA00) with the group-SPECIAL attribute.</td>
</tr>
<tr>
<td></td>
<td>User SPB1 wants to issue the command as a RACF operator command, and the RACF subsystem prefix is @.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Command</th>
<th>@DELUSER CA00</th>
</tr>
</thead>
<tbody>
<tr>
<td>Defaults</td>
<td>None.</td>
</tr>
</tbody>
</table>
DISPLAY

DISPLAY (Display signed-on-from list)

Background
Persistent verification allows users to sign on to a partner LU (logical unit) and have their authority persist. In other words, once a user has signed on, a password is not required for subsequent signon attempts.

APPC/MVS invokes RACF to create and maintain a list called the signed-on-from list. If persistent verification is being used, the signed-on-from list consists of the users currently signed on with persistent verification authority.

Purpose
The RACF DISPLAY operator command displays information held in the signed-on-from list. Entries in the signed-on-from list possess the following information:

- User ID
- Group
- APPL (the local LU name)
- POE (the partner LU name from which the user is signed on)
- SECLABEL

The DISPLAY command has operands which correspond to the items listed above. You can use these operands to select which user entries to display from the signed-on-from list.

The information is displayed as a list of entries sorted by local LU. If there are multiple entries for a given local LU, these entries are sorted by user ID.

Issuing options

The following table identifies the eligible options for issuing the DISPLAY command:

<table>
<thead>
<tr>
<th>As a RACF TSO command?</th>
<th>As a RACF operator command?</th>
<th>With command direction?</th>
<th>With automatic command direction?</th>
<th>From the RACF parameter library?</th>
</tr>
</thead>
<tbody>
<tr>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>

For information on issuing this command as a RACF operator command, see "Rules for entering RACF operator commands" on page 21. You must be logged on to the console to issue this command as a RACF operator command.

Related commands
Use the SIGNOFF command to remove users from the signed-on-from list.

Authorization required
When issuing the DISPLAY command as a RACF operator command, you might require sufficient authority to the proper resource in the OPERCMDS class. See z/OS Security Server RACF Security Administrator's Guide for further information.
**Syntax**

For the key to the symbols used in the command syntax diagrams, see "Syntax of RACF commands and operands" on page 9. The complete syntax of the DISPLAY command is:

```
subsystem-prefix DISPLAY
    [ SIGNON ]
    [ APPL(local-luname | * ) ]
    [ POE(partner-luname | * ) ]
    [ USER(userid-name | * ) ]
    [ GROUP(group-name | * ) ]
    [ SECLABEL(security-label | * ) ]
```

For information on issuing this command as a RACF operator command, see "Rules for entering RACF operator commands" on page 21. You must be logged on to the console to issue this command as a RACF operator command.

**Parameters**

**subsystem-prefix**

The subsystem prefix identifies that the RACF subsystem is the processing environment. The subsystem prefix can be either the installation-defined prefix for RACF (1–8 characters) or, if no prefix has been defined, the RACF subsystem name followed by a blank. If the command prefix was registered with CPF, you can use the MVS command D OPDATA to display it or you can contact your RACF security administrator.

Only specify the subsystem prefix when issuing this command as a RACF operator command. The subsystem prefix is required when issuing RACF operator commands.

**SIGNON**

This operand indicates that the information to be displayed is from the signed-on-from list. Because this is always the case, this operand is a default value and can be omitted from the command line.

The operands below allow the operator to select the necessary search criteria. These operands are all optional.

- If none of the operands are specified, you receive an informational message, indicating the version, release and modification level for RACF.
- If no local LU is currently active, you receive an informational message.
- If you specify the APPL operand and at least one local LU is currently active, you receive an informational message with the names of the LU applications listed.

**APPL(local-luname | *)**

The `local-luname` is a 1–8 character name of the local LU to be searched for. An asterisk can occupy the last position of the name in order to provide a partial generic selection capability. A character string consisting of a single asterisk is permitted as a full generic that matches any APPL name in the signed-on-from list. A single asterisk is the default value.

**POE(partner-luname | *)**

The `partner-luname` is the name of the partner LU to be searched for. It can be a 1–8 character unqualified LU name or a 1–17 character network qualified LU name in the format `netid.luname`, where `netid` and `luname` are each 1–8 characters. If the netid is omitted, all partner LUs with the specified LU name
portion is shown (POE(LU1) would show NET1.LU1 and NET2.LU1). An asterisk can occupy the last position of the partner-luname in order to provide a partial generic selection capability. For example, the partner-luname NW1.LU2 would match with *, N*, NW*, NW1*, NW1.L*, NW1.LU*, NW1.LU2*, L*, LU*, and LU2*. A character string consisting of a single asterisk is permitted as a full generic that matches any POE name in the signed-on-from list. A single asterisk is the default if another operand (besides SIGNON) is specified.

**USER(userid-name | *)**

The userid-name is a 1–8 character name that represents the RACF user ID to be searched for. An asterisk can occupy the last position of the userid-name in order to provide a partial generic selection capability. A character string consisting of a single asterisk is permitted as a full generic that matches any user ID in the signed-on-from list. A single asterisk is the default value if either the GROUP operand or the SECLABEL operand is specified.

**GROUP(group-name | *)**

The group-name is a 1–8 character name of the RACF group to be searched for. An asterisk can occupy the last position of the group-name in order to provide a partial generic selection capability. A character string consisting of a single asterisk is permitted as a full generic that matches any group name in the signed-on-from list. A single asterisk is the default value if either the USER operand or the SECLABEL operand is specified. Note that entries in the signed-on-from list might not always be added to that list with a group-name value. Such entries have group-name values consisting of blanks.

**SECLABEL(security-label | *)**

The security-label is a 1–8 character name which represents the RACF security label to be searched for. An asterisk can occupy the last position of the specification in order to provide a partial generic selection capability. A character string consisting of a single asterisk is permitted as a full generic that matches any security label in the signed-on-from list. A single asterisk is the default value if either the USER operand or the GROUP operand is specified.

**Examples**

**Table 13. DISPLAY examples**

**Example 1**

**Operation** Display all the partner LUs associated with a particular local LU.

**Known** The local LU name is locallu. The RACF subsystem prefix is @.

**Command** @display appl(locallu),poe(*)

**Defaults** SIGNON

**Output** See Figure 4

**Example 2**

**Operation** Display all the users signed on for a particular LU pair.

**Known** The local LU is locallu, the partner LU is prtnrlu1. The RACF subsystem prefix is @.

**Command** @display appl(locallu),poe(prtnrlu1),user(*)

**Defaults** SIGNON, GROUP(*), and SECLABEL(*)

**Output** See Figure 3

**Example 3**

**Operation** Display each local LU and its associated partner LUs, and for each LU pair, display the users signed on.

**Known** The RACF subsystem prefix is @.

**Command** @display appl(*),poe(*),user(*)

**Defaults** SIGNON, GROUP(*), and SECLABEL(*)

**Output** See Figure 6

**Important:** In many instances, this command might generate large amounts of display output.
**Example 4**

**Operation**  
Display each local LU and its associated partner LUs, and for each LU pair, display the users with `userid_names` beginning with `B`.

**Known**  
The RACF subsystem prefix is `@`.

**Command**  
```
@display appl(*),poe(*),user(B*),group(*)
```

**Defaults**  
SIGNON and SECLABEL(*)

**Output**  
See Figure 7.

**Example 5**

**Operation**  
Display all the LU pairs that users have signed on to using a particular group.

**Known**  
The RACF subsystem prefix is `@`. The `group-name` is `grp1`.

**Command**  
```
@display group(grp1),appl(*),poe(*),user(*)
```

**Defaults**  
SIGNON, SECLABEL(*)

**Output**  
See Figure 8.
DISPLAY

Figure 7. Example 4: Output for the DISPLAY Command

Figure 8. Example 5: Output for the DISPLAY Command
HELP (Obtain RACF help)

Purpose

Use the HELP command to obtain information about the function, syntax, and operands of RACF TSO commands. This information is displayed at your terminal in response to your request for help.

Note: When you use the HELP command to display the syntax for a RACF command, the brackets and braces shown in the syntax diagrams in this book are not displayed on your terminal, and a blank can appear in place of a bracket or brace. If you are unsure whether an operand is optional or required, you should refer to the syntax diagrams contained in this book.

Authorization required

You need no special attribute or authority to use the HELP command. Any user who can log on to TSO can issue this command.

Issuing options

The following table identifies the eligible options for issuing the HELP command:

<table>
<thead>
<tr>
<th>As a RACF TSO command?</th>
<th>As a RACF operator command?</th>
<th>With command direction?</th>
<th>With automatic command direction?</th>
<th>From the RACF parameter library?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
</tbody>
</table>

For information on issuing this command as a RACF TSO command, refer to Chapter 3, “RACF TSO commands,” on page 15.

Syntax

For the key to the symbols used in the command syntax diagrams, see “Syntax of RACF commands and operands” on page 9. The complete syntax of the HELP command is:

```plaintext
{ HELP | H }
  [ command-name ]
  [ ALL ]
  [ FUNCTION ]
  [ OPERANDS [ (operand ...)] ]
  [ SYNTAX ]
```

Parameters

`command-name`

Specifies the name of the command about which you want information.

If you specify this operand, it must be the first operand following HELP. If you omit this operand, you obtain a list of all the TSO commands. However, this list does not contain any RACF commands. Therefore, if you want information about a particular RACF command, you must specify the RACF command name following HELP. Help information is available for all RACF commands that are documented in this book.
HELP

**ALL**
Specifies that you want to see all the available information about the command.
This information includes the function, syntax, and operands of the command. If
no other operand is specified, ALL is the default value.

**FUNCTION**
Specifies that you want to see information about the purpose and operation of
the command.

**OPERANDS**
Specifies that you want to see information about the operands of the command.
When you specify OPERANDS and omit any values, all operands for the
command is described. To obtain information about a particular operand, specify
that operand within parentheses following OPERANDS. If you specify more than
one operand, separate the operand names by either commas or blanks.

**SYNTAX**
Specifies that you want to see information about the proper syntax of the
command.

**Examples**

<table>
<thead>
<tr>
<th>Example</th>
<th>Operation</th>
<th>Command</th>
<th>Known</th>
<th>Defaults</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example 1</td>
<td>User LQJ0 wants to see all available information for the ADDUSER command. Known User LQJ0 is RACF-defined.</td>
<td>HELP ADDUSER</td>
<td>ALL</td>
<td></td>
</tr>
<tr>
<td>Example 2</td>
<td>User JXN01 wants to see a description of the AUDIT, ADDMEM, DATA, and SECLEVEL operands for the RDEFINE command. Known User JXN01 is RACF-defined.</td>
<td>HELP RDEFINE OPERANDS(AUDIT ADDMEM DATA SECLEVEL)</td>
<td>None.</td>
<td></td>
</tr>
<tr>
<td>Example 3</td>
<td>User MJW02 wants to see a description of the function and syntax of the SETROPTS command. Known User MJW02 is RACF-defined.</td>
<td>HELP SETROPTS FUNCTION SYNTAX</td>
<td>None.</td>
<td></td>
</tr>
</tbody>
</table>
LISTDSD (List data set profile)

Purpose

Use the LISTDSD command to list information included in tape and DASD data set profiles. A data set profile consists of a RACF segment and, optionally, a DFP or TME segment. The LISTDSD command provides you with the option of listing information contained in the entire data set profile (all segments), or listing the information contained only in a specific segment of the profile.

You can request the details for any number of profiles by giving the full name of each profile. You can also request the details for all profiles whose names are qualified by specific user IDs, group names, or character strings.

You can use the LISTDSD command to cause the changes to go into effect for the generic profiles after issuing the ADDSD, ALTDSMD, or DELDSD commands. LISTDSD places a new copy of the profile in the user's address space.

Details RACF lists from the RACF segment of each profile:

- The level
- The owner
- The type of access attempts (as specified by the AUDIT operand on the ADDSD or ALTDSMD command) that are being logged on the SMF data set
- The universal access authority
- Your highest level of access authority
- The group under which the profile was created
- The data set type (tape, VSAM, non-VSAM, or MODEL)
- The retention period for a tape data set
- The type of access attempts (as specified by the GLOBALAUDIT operand on the ALTDSMD command) that are being logged on the SMF data set (for auditors only)
- The volume serial number (volser) of the volume on which the data set resides.
  For both a single volume and multivolume VSAM data set, the volser represents the volume containing the catalog entry for the data set.
  For a non-VSAM data set, the volser represents the volume containing the data set itself. If it is a multivolume non-VSAM data set, a list of volsers is given. The list represents the volumes on which the protected data set resides. They are listed in the order in which they were defined.
- Unit information for the data set (if unit information had been specified in the UNIT operand on the ADDSD or ALTDSMD command)
- Installation-defined data as specified on the DATA operand of the ADDSD or ALTDSMD command.

Note: If your installation is running with maximum security (that is, with SETROPTS MLSTABLE, MLS, MLACTIVE, and SECLABELCONTROL all active and the SECLABEL class active), this information is listed only for those with SPECIAL. If you are not SPECIAL, the following text appears in your output in the installation data field: * SUPPRESSED *

Additional details listed: You can request the following additional details by using the appropriate LISTDSD operands:

- Historical data, such as the date the data set was:
  - Defined to RACF
LISTDSD

- Last referenced
- Last updated
For additional information, see the HISTORY operand.

- The number of times the data set was accessed by all users for each of the following access authorities:
  - ALTER, CONTROL, UPDATE, READ, EXECUTE.
For additional information, see the STATISTICS operand.

**Note:** These details are not meaningful if resource statistics gathering is bypassed at your installation. For a generic profile, RACF replaces any statistics line with NOT APPLICABLE FOR GENERIC PROFILE.

- The standard access list, which displays:
  - All users and groups authorized to access the data set
  - The level of authority for each user and group
  - The number of times each user has accessed the data set
For additional information, see the AUTHUSER operand.

- The conditional access list, which displays the same fields as the standard access list as well as the following fields:
  - The class of the resource
  - The entity name of the resource
For additional information, see the AUTHUSER operand.

- The information listed below:
  - The user categories authorized to access the data set
  - The security level required to access the data set
  - The security label required to access the data set
For additional information, see the AUTHUSER operand.

- The details RACF lists from the DFP segment of the profile:
  - The user ID or group name of the data set resource owner
- The details RACF lists from the TME segment of the profile:
  - The roles and associated access levels

**RACF date handling:** RACF interprets dates with 2-digit years as follows. (The \(yy\) value represents the 2-digit year.)

- If \(70 < yy \leq 99\), the date is interpreted as 19yy.
- If \(00 \leq yy < 70\), the date is interpreted as 20yy.

### Issuing options

The following table identifies the eligible options for issuing the LISTDSD command:

<table>
<thead>
<tr>
<th>As a RACF TSO command?</th>
<th>As a RACF operator command?</th>
<th>With command direction?</th>
<th>With automatic command direction?</th>
<th>From the RACF parameter library?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>

For information on issuing this command as a RACF TSO command, refer to Chapter 3, “RACF TSO commands,” on page 15.
LISTDSD

For information on issuing this command as a RACF operator command, refer to Chapter 4, “RACF operator commands,” on page 21.

You must be logged on to the console to issue this command as a RACF operator command.

Related commands

- To list a general resource profile, see RLIST (List general resource profile). (General resources include terminals and other resources defined in the class descriptor table.)
- To list a user profile, see LISTUSER (List user profile).
- To list a group profile, see LISTGRP (List group profile).
- To obtain a list of data set profiles, see SEARCH (Search RACF database).

Authorization required

When issuing the LISTDSD command as a RACF operator command, you might require sufficient authority to the proper resource in the OPERCMDS class. See z/OS Security Server RACF Security Administrator's Guide for further information.

To specify the AT keyword, you must have READ authority to the DIRECT.node resource in the RRSFDATA class and a user ID association must be established between the specified node.userid pair(s).

To specify the ONLYAT keyword you must have the SPECIAL attribute, the userid specified on the ONLYAT keyword must have the SPECIAL attribute, and a user ID association must be established between the specified node.userid pair(s) if the user IDs are not identical.

Listing the RACF segment of a data set profile: To list the details of the RACF segment of a data set profile, you must have a sufficient level of authority for each profile to be listed. One of the following conditions must be met for each profile to be listed:

- You have the SPECIAL attribute.
- The profile is within the scope of a group in which you have the group-SPECIAL attribute.
- You have the OPERATIONS attribute.
- The profile is within the scope of a group in which you have the group-OPERATIONS attribute.
- The high-level qualifier of the profile name (or the qualifier supplied by a command installation exit) is your user ID.
- You are the owner of the profile.
- You are on the profile's access list with at least READ authority. (If your level of authority is NONE, the data set is not listed.)
- Your current connect group (or, if list-of-groups checking is active, any group to which you are connected) is in the access list and has at least READ authority.
- The universal access authority is at least READ.
- You have at least READ access for the profile name from the GLOBAL ENTRY TABLE (if this table contains an entry for the profile).
- You have the AUDITOR attribute.
- The data set profile is within the scope of a group in which you have the group-AUDITOR attribute.
LISTDSD

To display the type of access attempts (as specified by the GLOBALAUDIT operand on the ALTDSD command) that are being logged on the SMF data set, either you must have the AUDITOR attribute or the profile must be within the scope of a group in which you have the group-AUDITOR attribute.

To specify the AUTHUSER operand to display the access list for a profile, one of the following conditions must be met for each profile to be listed:

- You have the SPECIAL attribute.
- You have the OPERATIONS attribute.
- You have the AUDITOR attribute.
- The profile is within the scope of a group in which you have the group-SPECIAL attribute.
- The profile is within the scope of a group in which you have the group-OPERATIONS attribute.
- The data set profile is within the scope of a group in which you have the group-AUDITOR attribute.
- The high-level qualifier of the profile name (or the qualifier supplied by a command installation exit) is your user ID.
- You are the owner of the profile.
- You have ALTER access for the profile name from the GLOBAL ENTRY TABLE (if this table contains an entry for the profile).
- For a discrete profile, you are on the profile’s access list with ALTER authority. (If you have any other level of authority, you cannot use the operand.)
- For a discrete profile, your current connect group (or, if list-of-groups checking is active, any group to which you are connected) is in the access list and has ALTER authority.
- For a discrete profile, the universal access authority is ALTER.

Profiles that contain inactive security labels may not be listed if SETROPTS SECLBYSYSTEM is active because only users with SPECIAL or AUDITOR authority are allowed to view inactive security labels.

Listing the DFP or TME segment of a data set profile: To list information within the segment of a data set profile, one of the following conditions must be true:

- You have the SPECIAL or AUDITOR attribute.
- You have at least READ authority to the desired field within the segment through field-level access control.

Syntax

For the key to the symbols used in the command syntax diagrams, see “Syntax of RACF commands and operands” on page 9. The complete syntax of the LISTDSD command is:
LISTDSD

[subsystem-prefix][LISTDSD | LD]

[ ALL ]
[ AT([node].userid ...) | ONLYAT([node].userid ...) ]
[ AUTHUSER ]
[ (DATASET(profile-name ...)  
  | ID(name ...)  
  | PREFIX(char ...)) ]
[ DFP ]
[ DSNS ]
[ GENERIC | NOGENERIC ]
[ HISTORY ]
[ NORACF ]
[ STATISTICS ]
[ TME ]
[ VOLUME(volume-serial ...) ]

For information on issuing this command as a RACF TSO command, refer to Chapter 3, “RACF TSO commands,” on page 15.

For information on issuing this command as a RACF operator command, refer to Chapter 4, “RACF operator commands,” on page 21.

Parameters

**subsystem-prefix**

Specifies that the RACF subsystem is the processing environment of the command. The subsystem prefix can be either the installation-defined prefix for RACF (1–8 characters) or, if no prefix has been defined, the RACF subsystem name followed by a blank. If the command prefix was registered with CPF, you can use the MVS command D OPDATA to display it or you can contact your RACF security administrator.

Only specify the subsystem prefix when issuing this command as a RACF operator command. The subsystem prefix is required when issuing RACF operator commands.

**ALL**

 Specifies that you want RACF to display all information for each data set.

The access list is included only if you have sufficient authority to use the AUTHUSER operand (see Authorization required). The type of access attempts (as specified by the GLOBALAUDIT operand on the ALTDSD command) that are being logged on the SMF data set is included only if you have the AUDITOR or group-AUDITOR attribute.

The DFP and TME segments must be requested explicitly.

**AT | ONLYAT**

The AT and ONLYAT keywords are only valid when the command is issued as a RACF TSO command.

**AT([node].userid ...)**

Specifies that the command is to be directed to the node specified by **node**, where it runs under the authority of the user specified by **userid** in the RACF subsystem address space.

If **node** is not specified, the command is directed to the local node.
LISTDSD

ONLYAT([node].userid ..)
LISTDSD is not eligible for automatic command direction. If you specify the
ONLYAT keyword, the effect is the same as if you specified the AT keyword.

AUTHUSER
Specifies that you want the following information included in the output:
• The user categories authorized to access the data set
• The security level required to access the data set
• The security label required to access the data set
• The standard access list. This contains the following:
  – All users and groups authorized to access the data set
  – The level of authority for each user and group
  – The number of times each user has accessed the data set. This detail is
    only meaningful when your installation is gathering resource statistics.
    This detail is not included in the output for generic profiles.
• The conditional access list. This list consists of the same fields as in the
  standard access list, as well as the following fields:
  – The class of the resource through which each user and group can access
    the data set. For example, if a user can access the data set through
    terminal TERM01, then TERMINAL would be the class listed.
  – The entity name of the resource through which each user and group can
    access the data set. In the example above, TERM01 would be listed.
You must have sufficient authorization to use the AUTHUSER operand (see
Authorization required).

DATASET | ID | PREFIX

DATASET(profile-name ..)
Specifies the names of one or more data sets whose profiles RACF is to
list. If a specified name appears more than once in the RACF database,
LISTDSD displays information about all the profiles with that name to which
you have proper authority.
The data set name you specify must be enclosed in single quotation marks
unless it is your own data set.
Because RACF uses the RACF database and not the catalog when
searching for data set profiles, you cannot use alias data set names.
Note that if you are using naming convention processing, either through the
naming convention table or an exit, the name you type might not be the
same as the name that appears in the output.

ID(name ..)
Specifies one or more user IDs or group names. All users and groups must
be defined to RACF. Details are listed for all discrete and generic profiles
that have the specified user IDs or group names as the high-level qualifier
name (or as the qualifier supplied by a command installation exit).
If you do not specify DATASET, PREFIX, or ID, RACF uses your user ID as
the default value for the ID operand.

PREFIX(char ..)
Specifies one or more character strings. Details are listed for all profiles
whose names begin with the specified character strings.
Note that comparison between the character strings and the profile names is not limited to the high-level qualifier. For example, if you specify PREFIX(A.B.C), RACF would display information for profiles such as A.B.C, A.B.CAD, and A.B.C.X.

**DFP**

Specifies that for a DFP-managed data set, you want to list the user ID or group name designated as the data set resource owner. (The data set resource owner, or RESOWNER, is distinguished from the OWNER, which represents the user or group that owns the data set profile.)

**DSNS**

Specifies that you want to list the cataloged data sets protected by the profile specified by the DATASET, ID, or PREFIX operand.

Only data sets cataloged in an Integrated Catalog Facility (ICF) catalog are listed.

Affected tape data sets are listed, regardless of what is specified for SETROPTS TAPEDSN, or whether the TAPEVOL class is active.

When data and index components of VSAM clusters are listed, they are followed by (D) or (I), respectively.

This operand might give unpredictable results if one of the following is true:

- You are using naming convention processing, either through the naming convention table or an exit, to modify data set names so they are protected by different profiles.
- You are using the PREFIX operand of SETROPTS to provide a high-level qualifier for data sets that have only one level in their names.
- There are migrated items in the list and either information about the item cannot be obtained from the migration facility or the migration facility is not available.

In these cases, RACF cannot verify that the item is protected by the input profile and the migrated item is included in the list and is followed by the ? character. Whenever these items are included in the list, the following message appears at the end of the list to explain the ? character.

**Notes:**

1. If a migrated cluster name appears in the list, but it has an alternate index or path, information on its data or index names is unavailable without recalling the cluster. This message appears after the cluster name:
   
   ** Migrated cluster component information
   ** not available without recall.

2. If a migrated cluster name appears in the list and LISTDSD cannot obtain the index and data names due to a migration facility error, this message appears after the cluster name:

   ** Migrated cluster component information
   ** not available.

3. If the name of a non-migrated cluster appears in the list and RACF is unable to obtain the data and index names specifically through this item, this message appears after the cluster name:

   ** Cluster component information
   ** not available.
LISTDSD

4. If the LISTDSD processor could not obtain all the information on one of the data sets potentially protected by the input profile, it includes the data set in the command output, but follow it with this message:

** Data set information not available. (x)

It is likely that this condition occurred because the data set was deleted between the time the LISTDSD DSNS processor first found the names of all the data sets potentially protected by the input profile and the time it processed that particular data set. If that is the case, ignore that data set entry. If that is not the case, issue the LISTDSD command again and if the additional message still appears, contact IBM support; (x) is a numeric value that denotes diagnostic information used by IBM support.

5. The LISTDSD command processor does not include the following items in the output list of protected data sets:
   - master catalog
   - alternate index (AIX®) and its components
   - catalogs

**GENERIC | NOGENERIC**

**GENERIC**

Specifies that RACF is to list only information for the generic profiles. If you specify GENERIC with DATASET, RACF lists information for generic profiles whose names most closely match the data set names you specify.

GENERIC, when specified with DATASET, causes changes to take effect after adding, changing, or deleting generic profiles. It places a fresh copy of the profile in the command user's address space.

**NOGENERIC**

Specifies that RACF is to list only information for discrete profiles.

**Notes:**

1. If you specify ID or PREFIX but omit GENERIC and NOGENERIC, RACF lists information for all discrete and generic profiles of the data sets owned or represented by the names specified in the command.

   For example, if you enter the following command:
   
   LISTDSD ID(SMITH)

   RACF lists all data set profiles for user ID SMITH.

2. If you specify the DATASET operand but omit GENERIC and NOGENERIC, RACF lists information for the discrete profile, if it exists, and the fully qualified generic profile if it exists, or the generic profile that is not fully qualified, if its name, including all its qualifiers, matches the name specified on the command.

   For example, if you enter the following command:
   
   LISTDSD DATASET('XXX.YYY', 'AA.*')

   RACF lists information for the discrete profile XXX.YYY, if it exists, the fully qualified generic profile XXX.YYY if it exists, and the generic profile AA.* if it exists.

3. If you specify DATASET with a fully qualified name for a data set that is protected by a generic profile that is not fully qualified, information for this profile can be listed only when GENERIC is specified.

   If you specified DATASET without GENERIC and NOGENERIC and you received an informational message (No RACF description found.) for one of
the specified fully qualified names, you might want to retry the command on
this name using GENERIC, because it is possible that this data set is
protected by a generic profile that is not fully qualified.
For example, data set BBB.CCC is protected by a generic profile BBB.*. If
you enter the following command:

\texttt{LISTDSD \textquoteleft \textquoteleft BBB.CCC\textquoteright \textquoteright}

RACF lists information only if there is a discrete profile BBB.CCC, or a fully
qualified generic profile BBB.CCC, or both. But if you enter the following
command:

\texttt{LISTDSD \textquoteleft \textquoteleft BBB.CCC\textquoteright \textquoteright \ \texttt{GENERIC}}

RACF lists information for the fully qualified generic profile BBB.CCC if it
exists, or the generic profile that most closely matches BBB.CCC. In this
example, the generic profile BBB.* is listed.

4. If generic profile command processing is inactive, only discrete profiles are
listed. RACF does not search for generic profiles.

\textbf{HISTORY}

\textit{Specifies that you want to list the following data:}
\begin{itemize}
  \item The date each profile was defined to RACF
  \item The date each data set was last referenced
  \item The date of the last authorization check for UPDATE authority
\end{itemize}

\textbf{NORACF}

\textit{Specifies that you want to suppress the listing of RACF segment information
from the specified data set's profile. If you specify NORACF, you must include
one or more of the following operands: DSNS, DFP, TME.}

If you do not specify NORACF, RACF displays the information in the RACF
segment of a data set.

The information displayed as a result of using the NORACF operand is
dependent on other operands used in the command. For example, if you use
NORACF with DSNS, DFP, or TME also specified, only that information (DSNS,
DFP, or TME) is displayed.

\textbf{STATISTICS}

\textit{Specifies that you want to list the statistics for each profile. The list includes the
number of times the profile was accessed by users with READ, UPDATE,
CONTROL, and ALTER authorities, as well as a separate total for each
authority level. These details are meaningful only when your installation is
gathering resource statistics. For generic profiles, RACF replaces any statistics
line with NOT APPLICABLE FOR GENERIC PROFILE.}

\textbf{TME}

\textit{Specifies that information for the Tivoli Security Management Application is to
be listed.}

\textbf{VOLUME(}volume-serial ...\textbf{)}

\textit{Limits the profiles listed to those found on the specific volume or list of volumes
identified by volume serial number. RACF does not list profiles with the same
name found on other volumes. If you do not specify NOGENERIC, RACF lists
any generic profiles as well.
### Examples

#### Table 15. LISTDSD examples

<table>
<thead>
<tr>
<th>Example 1</th>
<th>Operation</th>
<th>User DAF0 wants to list all information for his own data set profiles.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Known</td>
<td>User DAF0 is RACF-defined, and does not have the AUDITOR attribute. User DAF0 wants to issue the command as a RACF TSO command.</td>
</tr>
<tr>
<td></td>
<td>Command</td>
<td>LISTDSD ALL</td>
</tr>
<tr>
<td></td>
<td>Defaults</td>
<td>ID(DAF0)</td>
</tr>
<tr>
<td></td>
<td>Output</td>
<td>See <a href="#">Figure 9</a></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Example 2</th>
<th>Operation</th>
<th>User IA0 wants to list the users authorized to data set SYS1.PLIBASE.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Known</td>
<td>User IA0 has ALTER authority to SYS1.PLIBASE, and does not have the AUDITOR attribute. User IA0 wants to issue the command as a RACF TSO command.</td>
</tr>
<tr>
<td></td>
<td>Command</td>
<td>LISTDSD DATASET('SYS1.PLIBASE') AUTHUSER</td>
</tr>
<tr>
<td></td>
<td>Defaults</td>
<td>None.</td>
</tr>
<tr>
<td></td>
<td>Output</td>
<td>See <a href="#">Figure 10</a></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Example 3</th>
<th>Operation</th>
<th>User ADM1 wants to list a generic profile SALES.*.ABC.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Known</td>
<td>User ADM1 is the owner of the generic profile, and generic profile command processing is enabled. User ADM1 has the group-AUDITOR attribute in group SALES. User ADM1 wants to issue the command as a RACF TSO command.</td>
</tr>
<tr>
<td></td>
<td>Command</td>
<td>LISTDSD DATASET('SALES.*.ABC')</td>
</tr>
<tr>
<td></td>
<td>Defaults</td>
<td>None.</td>
</tr>
<tr>
<td></td>
<td>Output</td>
<td>See <a href="#">Figure 11</a></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Example 4</th>
<th>Operation</th>
<th>User JADAMS wants to display the discrete profile for the DFP-managed data set RESEARCH.TEST.DATA. JADAMS also wants to display the user or group who is the data set resource owner.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Known</td>
<td>User JADAMS is the owner of the profile protecting data set RESEARCH.TEST.DATA. User JADAMS has field-level access of at least READ for the DFP segment. User JADAMS wants to issue the command as a RACF TSO command.</td>
</tr>
<tr>
<td></td>
<td>Command</td>
<td>LISTDSD DATASET('RESEARCH.TEST.DATA') DFP</td>
</tr>
<tr>
<td></td>
<td>Defaults</td>
<td>None.</td>
</tr>
<tr>
<td></td>
<td>Output</td>
<td>See <a href="#">Figure 12</a></td>
</tr>
</tbody>
</table>
### LISTDSD Command Output

```
LISTDSD ALL
INFORMATION FOR DATASET DAF0.DS2.DATA
LEVEL  OWNER  UNIVERSAL ACCESS  WARNING  ERASE
--------  -------  ----------------  -------  -------
00  DAF0  READ  NO  NO
AUDITING
--------
SUCCESS(READ),FAILURES(ALTER)
NOTIFY
--------
NO USER TO BE NOTIFIED
YOUR ACCESS  CREATION GROUP  DATASET TYPE
--------------  --------------  --------------
NONE GIVEN  RESEARCH  NON-VSAM
VOLUMES ON WHICH DATASET RESIDES  UNIT
-------------------------------- ----
231406  SYSDA
NO INSTALLATION DATA
SECURITY LEVEL
-------------------------------------------
NO SECURITY LEVEL
CATEGORIES
----------
NOCATEGORIES
SECLABEL
----------
NO SECLABEL
CREATION DATE  LAST REFERENCE DATE  LAST CHANGE DATE
(DAY) (YEAR) (DAY) (YEAR) (DAY) (YEAR)
-------------------  -------------------  -------------------
145  85  145  85  145  85
ALTER COUNT  CONTROL COUNT  UPDATE COUNT  READ COUNT
-----------  -------------  ------------  ----------
00000  00010  00000  00010
ID  ACCESS  ACCESS COUNT
------  -------  ******
IA0  READ  00010
ADM1  READ  00000
PROJECTA  UPDATE  00008
ID  ACCESS  ACCESS COUNT  CLASS  ENTITY NAME
------  -------  ******  -----  ******
NO ENTRIES IN CONDITIONAL ACCESS LIST
```

Figure 9. Example 1: Output for the LISTDSD Command (Part 1 of 2)
Figure 9. Example 1: Output for the LISTDSD Command (Part 2 of 2)
### LISTDSD Command Output Examples

#### Example 2: Output for the LISTDSD Command

```
LISTDSD DATASET('SYS1.PLIBASE') AUTHUSER
INFORMATION FOR DATASET SYS1.PLIBASE
LEVEL OWNER UNIVERSAL ACCESS WARNING ERASE
----- -------- ---------------- ------- -----
00 IAO READ NO NO

AUDITING

SUCCESS(UPDATE)
NOTIFY

---

NO USER TO BE NOTIFIED

YOUR ACCESS CREATION GROUP DATASET TYPE
----------- -------------- ------------
ALTER SYS1 NON-VSAM

VOLUMES ON WHICH DATASET RESIDES UNIT
--------------------------------- ----
231407 SYSDA

INSTALLATION DATA

PL/1 LINK LIBRARY

SECURITY LEVEL

---

NO SECURITY LEVEL
CATEGORIES

---

NOCATEGORIES
SECLABEL

---

NO SECLABEL

ID ACCESS ACCESS COUNT
------- ------ ------------
ESH25 UPDATE 00009
PROJECTB READ 00015
IA0 ALTER 00020

ID ACCESS ACCESS COUNT CLASS ENTITY NAME
------- ------ ------------ ----- -----------

NO ENTRIES IN CONDITIONAL ACCESS LIST
```

**Figure 10. Example 2: Output for the LISTDSD Command**

#### Example 3: Output for the LISTDSD Command

```
LISTDSD DATASET('SALES.*.ABC')
INFORMATION FOR DATASET SALES.*.ABC (G)
LEVEL OWNER UNIVERSAL ACCESS WARNING ERASE
----- -------- ---------------- ------- -----
00 ADM1 READ NO NO

AUDITING

ALL(READ)
NOTIFY

---

NO USER TO BE NOTIFIED

YOUR ACCESS CREATION GROUP DATASET TYPE
----------- -------------- ------------
NONE GIVEN RESEARCH NON-VSAM

GLOBALAUDIT

---

NONE

NO INSTALLATION DATA
```

**Figure 11. Example 3: Output for the LISTDSD Command**
### LISTDSD

```plaintext
LISTDSD DATASET('RESEARCH.TEST.DATA') DFP
INFORMATION FOR DATASET RESEARCH.TEST.DATA
LEVEL OWNER UNIVERSAL ACCESS WARNING ERASE
----- -------- ---------------- ------- -----
00 JADAMS READ NO NO
AUDITING
--------
ALL(READ)
NOTIFY
--------
NO USER TO BE NOTIFIED
YOUR ACCESS CREATION GROUP DATASET TYPE
----------- -------------- ------------
NONE GIVEN RESEARCH NON-VSAM
GLOBALAUDIT
-----------
NONE
NO INSTALLATION DATA
DFP INFORMATION
-----------
REOWNER= KSMITH
```

Figure 12. Example 4: Output for the LISTDSD Command
LISTGRP (List group profile)

Purpose

Use the LISTGRP command to list details of specific RACF group profiles. A group profile consists of a RACF segment and, optionally, other segments such as DFP and OMVS. The LISTGRP command provides you with the option of listing the information contained in the entire group profile (all segments), or listing the information contained only in a specific segment of the group profile.

The details RACF lists from the RACF segment of each group profile are:
- The superior group of the group
- The owner of the group
- The date the group was defined to RACF
- The terminal option of the group
- Whether or not the group is a universal group
- Any subgroups under the group
- Installation-defined data, as specified by the DATA operand of the ADDGROUP and ALTGROUP command
- The name of the data set model profile.

RACF lists the following information from the RACF segment of the group profile for each user connected to the group:
- The user ID
- An exception to this is when the group is a UNIVERSAL group. When a UNIVERSAL group displayed with the LISTGRP command, not all members will be listed. Only users with authority higher than USE or with the attributes SPECIAL, OPERATIONS or AUDITOR at the group level will be shown in the member list. To view all members of a UNIVERSAL group, the Database Unload Utility (IRRDBU00) must be used. For more information on using the Database Utility (IRRDBU00), see z/OS Security Server RACF Security Administrator’s Guide.
- The user’s level of authority in the group
- The number of times the user has entered the system using this group as the current connect group
- The user’s default universal access authority
- The user’s connect attributes (group-related user attributes)
- Any REVOKE or RESUME processing either in effect or pending, with the corresponding dates even if they have passed.

The details RACF lists from the DFP segment of the group profile are:
- The group’s default data class
- The group’s default management class
- The group’s default storage class
- The data management data application for the group.

The details RACF lists from the TME segment of the group profile are:
- The list of roles that refer to this group.

The details RACF lists from the OMVS or OVM segment of the group profile are:
- The group’s z/OS UNIX System Services group identifier.
LISTGRP

The details RACF lists from the CSDATA segment of the group profile are:
- The list of custom fields that your installation has added to this group.

Issuing options

The following table identifies the eligible options for issuing the LISTGRP command:

<table>
<thead>
<tr>
<th>As a RACF TSO command?</th>
<th>As a RACF operator command?</th>
<th>With command direction?</th>
<th>With automatic command direction?</th>
<th>From the RACF parameter library?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>

For information on issuing this command as a RACF TSO command, refer to Chapter 3, “RACF TSO commands,” on page 15.

For information on issuing this command as a RACF operator command, refer to Chapter 4, “RACF operator commands,” on page 21.

You must be logged on to the console to issue this command as a RACF operator command.

Related commands

- To list a user profile, see LISTUSER (List user profile).
- To list a data set profile, see LISTDSD (List data set profile).
- To list a general resource profile, see RLIST (List general resource profile).
  (General resources include terminals and other resources defined in the class descriptor table.)
- To obtain a list of group profiles, see SEARCH (Search RACF database).

Authorization required

When issuing the LISTGRP command as a RACF operator command, you might require sufficient authority to the proper resource in the OPERCMDS class. See z/OS Security Server RACF Security Administrator’s Guide for further information.

To specify the AT keyword, you must have READ authority to the DIRECT.node resource in the RRSFDATA class and a user ID association must be established between the specified node.userid pair(s).

To specify the ONLYAT keyword you must have the SPECIAL attribute, the userid specified on the ONLYAT keyword must have the SPECIAL attribute, and a user ID association must be established between the specified node.userid pair(s) if the user IDs are not identical.

**Listing the RACF segment of a group profile:** To list the details of the RACF segment of a group profile, one of the following conditions must be true:
- You have the SPECIAL attribute.
- You have the group-SPECIAL attribute in each group to be listed, or each group to be listed is within the scope of a group in which you have the group-SPECIAL attribute.
- You have the AUDITOR attribute.
- You have the group-AUDITOR attribute in each group to be listed, or each group to be listed is within the scope of a group in which you have the group-AUDITOR attribute.
LISTGRP

- You are the owner of the group.
- You have JOIN or CONNECT authority in the group.

**Listing the other segments of a group profile:** To list information from segments other than the RACF segment for a group profile, one of the following conditions must be true:
- You have the SPECIAL or AUDITOR attribute.
- You have at least READ authority to the desired field through field-level access control.

**Syntax**

For the key to the symbols used in the command syntax diagrams, see "Syntax of RACF commands and operands" on page 9. The complete syntax of the LISTGRP command is:

```
[subsystem-prefix][LISTGRP | LG]
[ ( [(group-name ... | *)] [ AT([node].userid ... | ONLYAT([node].userid ... ] ) [ CSDATA ] [ DFP ] [ NORACF ] [ OMVS ] [ OVM ] [ TME ]
```

For information on issuing this command as a RACF TSO command, refer to Chapter 3, “RACF TSO commands,” on page 15.

For information on issuing this command as a RACF operator command, refer to Chapter 4, “RACF operator commands,” on page 21.

**Parameters**

**subsystem-prefix**
- Specifies that the RACF subsystem is the processing environment of the command. The subsystem prefix can be either the installation-defined prefix for RACF (1–8 characters) or, if no prefix has been defined, the RACF subsystem name followed by a blank. If the command prefix was registered with CPF, you can use the MVS command D OPDATA to display it or you can contact your RACF security administrator.

Only specify the subsystem prefix when issuing this command as a RACF operator command. The subsystem prefix is required when issuing RACF operator commands.

**group-name | * **

- **group-name**
  - Specifies the name of one or more RACF-defined groups. If you specify more than one group name, you must enclose the names in parentheses.

- *** **
  - Specifies that you want to list information contained in all RACF-defined group profiles to which you have the required authority.

On a system with many groups defined, the use of * may result in a large amount of output and may not be useful to a user issuing the command. It
LISTGRP

may be more appropriate for the user to browse the output of IRRDBU00 (database unload utility) or to write a program to process the IRRDBU00 output and produce a report showing only the subset of information that is of interest to the user. The processing of output of LISTGRP by programs is not supported nor recommended by IBM. If you want a listing of all the groups for use by a program you should instead have the program process the output from IRRDBU00, RACROUTE REQUEST=EXTRACT, or ICHEINTY.

If you specify a group name or *, it must be the first operand following LISTGRP.

If you specify one or more group names (or *) without specifying an additional operand, RACF lists only the RACF segment information from the specified profiles.

If you enter LISTGRP with no operands, RACF lists only the RACF segment information from your current connect group.

AT | ONLYAT
The AT and ONLYAT keywords are valid only when the command is issued as a RACF TSO command.

AT([(node].userid ...)
Specifies that the command is to be directed to the node specified by node, where it runs under the authority of the user specified by userid in the RACF subsystem address space.

If node is not specified, the command is directed to the local node.

ONLYAT([(node].userid ...)
LISTGRP is not eligible for automatic command direction. If you specify the ONLYAT keyword, the effect is the same as if you specified the AT keyword.

CSDATA
Specifies that you want to list custom field information for this group. The custom field information in the CSDATA segment for this group was added using the ADDGROUP and ALTGROUP commands.

If you specify CSDATA you must also specify a group name or *.

Usage for each custom field is defined using the CFDEF operand of the RDEFINE command for resource profiles in the CFIELD class. Contact your security administrator to see how custom fields are used at your installation. For more information about custom fields, see z/OS Security Server RACF Security Administrator’s Guide.

DFP
Specifies that you want to list the information contained in the DFP segment of the group profile.

If you specify DFP you must also specify a group name or *.

NORACF
Specifies that you want to suppress the listing of base segment information from the group profile. If you specify NORACF, you must also specify one of the other segment names such as DFP or OMVS.

If you do not specify NORACF, RACF displays the information in the RACF segment of a group profile.
LISTGRP

OMVS
Specifies that you want to list the information contained in the OMVS segment of the group profile.

If you specify OMVS, you must also specify a group name or (*).

If the group profile contains an OMVS segment but GID was not specified on an ADDGROUP or ALTGROUP command, the listing displays the field name followed by the word NONE.

OVM
Specifies that you want to list the information contained in the OVM segment of the group profile.

If you specify OVM, you must also specify a group name or an (*).

If the group profile contains an OVM segment but GID was not specified on a ADDGROUP or ALTGROUP command, the listing displays the field name followed by the word NONE.

TME
Specifies that information for the Tivoli Security Management Application is to be listed.

If you specify TME you must also specify a group name or an asterisk (*).

Examples

Table 16. LISTGRP examples

Example 1
Operation User IA0 wants to display the information contained in the RACF segment of the profile for group RESEARCH.

Known User IA0 has CONNECT authority to group RESEARCH. User IA0 wants to issue the command as a RACF TSO command.

Command LISTGRP RESEARCH
Defaults None.

Output See Figure 13

Example 2
Operation User ADM1 wants to display the information contained in the RACF segment of the profiles for all groups.

Known User ADM1 has the SPECIAL and AUDITOR attributes. User ADM1 wants to issue the command as a RACF TSO command.

Command LISTGRP *
Defaults None.

Output See Figure 14

Example 3
Operation User ADM1 wants to display the information contained in the RACF segment and DFP segment of the profile for group DFPADMN.

Known User ADM1 has the SPECIAL and AUDITOR attributes.

Group DFPADMN is defined to RACF, and DFPADMN’s profile contains a DFP segment.

User ADM1 wants to issue the command as a RACF TSO command.

Command LISTGRP DFPADMN DFP
Defaults None.

Output See Figure 15
<table>
<thead>
<tr>
<th>Example 4</th>
<th>Operation</th>
<th>User ADM1 wants to display the information contained in only the DFP segment of the profile for group DFPADMN.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Known</td>
<td>User ADM1 has the SPECIAL and AUDITOR attributes.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Group DFPADMN is defined to RACF, and DFPADMN's profile contains a DFP segment.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>User ADM1 wants to issue the command as a RACF TSO command.</td>
<td></td>
</tr>
<tr>
<td>Command</td>
<td>LISTGRP DFPADMN DFP NORACF</td>
<td></td>
</tr>
<tr>
<td>Defaults</td>
<td>None.</td>
<td></td>
</tr>
<tr>
<td>Output</td>
<td>See Figure 16.</td>
<td></td>
</tr>
</tbody>
</table>

| Example 5 | Operation | User ADM1 requests the listing of the OMVS segment for the group OMVSG1.                              |
| Known     | User ADM1 has the SPECIAL attribute. User ADM1 wants to issue the command as a RACF TSO command.     |
| Command   | LISTGRP OMVSG1 OMVS NORACF                                                                           |
| Defaults  | None.                                                                                                  |
| Output    | See Figure 17.                                                                                        |

| Example 6 | Operation | User NETADM requests the listing of the UNIVERSAL group NETGROUP.                                     |
| Known     | NETGROUP is a UNIVERSAL group and only users with authority higher than USE or users with SPECIAL, OPERATIONS and AUDITOR at the GROUP level will be displayed in the member list. User NETADM has the SPECIAL attribute to the group NETGROUP. User NETADM wants to issue the command as a RACF TSO command. |
| Command   | LISTGRP NETGROUP                                                                                      |
| Defaults  | None.                                                                                                  |
| Output    | See Figure 18.                                                                                         |
Figure 13. Example 1: Output for LISTGRP RESEARCH

LISTGRP RESEARCH
INFORMATION FOR GROUP RESEARCH
SUPERIOR GROUP=SYS1 OWNER=IBMUSER CREATED=06.123
NO INSTALLATION DATA
NO MODEL DATA SET
TERMUACC
SUBGROUP(S)= PAYROLLB
USER(S)= ACCESS= ACCESS COUNT= UNIVERSAL ACCESS=
IBMUSER JOIN 000000 ALTER
    CONNECT ATTRIBUTES=None 
    REVOKE DATE=None 
    RESUME DATE=None 
DAF0 JOIN 000002 READ
    CONNECT ATTRIBUTES=None 
    REVOKE DATE=None 
    RESUME DATE=None 
IA0 CONNECT 000004 READ
    CONNECT ATTRIBUTES=ADSP SPECIAL OPERATIONS
    REVOKE DATE=None 
    RESUME DATE=None 
ESH25 USE 000000 READ
    CONNECT ATTRIBUTES=None 
    REVOKE DATE=None 
    RESUME DATE=None 
PROJECTB USE 000000 READ
    CONNECT ATTRIBUTES=None 
    REVOKE DATE=None 
    RESUME DATE=None 
RV2 CREATE 000000 READ
    CONNECT ATTRIBUTES=None 
    REVOKE DATE=None 
    RESUME DATE=None 
RV3 CREATE 000000 READ
    CONNECT ATTRIBUTES=None 
    REVOKE DATE=None 
    RESUME DATE=None 
ADM1 JOIN 000000 READ
    CONNECT ATTRIBUTES=OPERATIONS
    REVOKE DATE=None 
    RESUME DATE=None 
AEH0 USE 000000 READ
    CONNECT ATTRIBUTES=REVOKED
    REVOKE DATE=None 
    RESUME DATE=None
Figure 14. Example 2: Output for LISTGRP *
LISTGRP DFPADMN DFP
INFORMATION FOR GROUP DFPADMN
SUPERIOR GROUP=SYSADMN OWNER=SYSADMN CREATED=06.123
NO INSTALLATION DATA
NO MODEL DATA SET
TERMUACC
SUBGROUP(S)= DFPGRP1, DFPGRP2
USER(S)= ACCESS= ACCESS COUNT= UNIVERSAL ACCESS=
IBMUSER JOIN 000000 ALTER
    CONNECT ATTRIBUTES=NONE RESUME DATE=NONE
DSMITH JOIN 000002 READ
    CONNECT ATTRIBUTES=NONE RESUME DATE=NONE
HOTROD CONNECT 000004 READ
    CONNECT ATTRIBUTES=ADSP SPECIAL OPERATIONS
    REVOKE DATE=NONE RESUME DATE=NONE
ESHAW USE 000000 READ
    CONNECT ATTRIBUTES=NONE REVOKE DATE=NONE
    RESUME DATE=NONE
PROJECTB USE 000000 READ
    CONNECT ATTRIBUTES=NONE REVOKE DATE=NONE
    RESUME DATE=NONE
ADM1 JOIN 000000 READ
    CONNECT ATTRIBUTES=OPERATIONS REVOKE DATE=NONE
    RESUME DATE=NONE
AEHALL USE 000000 READ
    CONNECT ATTRIBUTES=REVOKED
    REVOKE DATE=NONE RESUME DATE=NONE
    DFP INFORMATION
    MGMTCLAS= DFP2MGMT
    STORCLAS= DFP2STOR
    DATACLAS= DFP2DATA
    DATAAPPL= DFP2APPL

Figure 15. Example 3: Output for LISTGRP DFPADMIN DFP

LISTGRP DFPADMN DFP NORACF
INFORMATION FOR GROUP DFPADMN
    DFP INFORMATION
    MGMTCLAS= DFP2MGMT
    STORCLAS= DFP2STOR
    DATACLAS= DFP2DATA
    DATAAPPL= DFP2APPL

Figure 16. Example 4: Output for LISTGRP DFPADMIN DFP NORACF

LISTGRP OMVSG1 OMVS NORACF
INFORMATION FOR GROUP OMVSG1
    OMVS INFORMATION
    GID= 0000003243

Figure 17. Example 5: Output for LISTGRP OMVSG1 OMVS NORACF
LISTGRP NETGROUP
INFORMATION FOR GROUP NETGROUP
  SUPERIOR GROUP=SYS1   OWNER=IBMUSER   CREATED=06.123
  NO INSTALLATION DATA
  NO MODEL DATA SET
  TERMUACC
  UNIVERSAL
  NO SUBGROUPS
  USER(S)= ACCESS= ACCESS COUNT= UNIVERSAL ACCESS=
          IBMUSER   JOIN  00000000   NONE
          CONNECT ATTRIBUTES= NONE
          REVOKE DATE= NONE   RESUME DATE= NONE
  NETADM CREATE  00000000   READ
          CONNECT ATTRIBUTES= SPECIAL
          REVOKE DATE= NONE   RESUME DATE= NONE

Figure 18. Example 6: Output for LISTGRP NETGROUP
LISTUSER

LISTUSER (List user profile)

Purpose

Use the LISTUSER command to list the details of specific RACF user profiles. A user profile consists of a RACF segment and, optionally, other segments such as TSO and DFP. The LISTUSER command provides you with the option of listing the information contained in the entire user profile (all segments), or listing the information contained only in specific segments of the user profile.

You cannot use the LISTUSER command to list information about user ID associations; you must use the RACLINK command.

The details RACF lists from the RACF segment for each user profile are:

- The user ID
- The user’s name or UNKNOWN, if the user's name was not specified on the ADDUSER command
- The owner of the user’s profile
- The date the user was defined to RACF
- The default group
- The date the user’s password was last updated
- The date the user’s password phrase was last updated
- The change interval (in number of days)
- Information about the user's password envelope and password phrase envelope, if any. (See “Details about listing envelope information” on page 238)
- The user’s attributes
- The date and time the user last entered the system
- The classes in which the user is authorized to define profiles
- The installation-defined data
  - If your z/OS installation is configured as a multilevel-secure environment, this information is not listed in your output. The output line * SUPPRESSED * appears under the installation data field. Only those with SPECIAL will be allowed to list the field.
- The name of default data set model profile
- Any REVOKE or RESUME processing either in effect or pending, with the corresponding dates even if they have passed
- The security label, the security level, and category
  - When you specify the user ID on the LISTUSER command, the default security label from the user profile in the RACF database is displayed in the output.
  - When you do not specify the user ID on the LISTUSER command, the security label you are currently logged on with (from the in-storage ACEE control blocks) is displayed in the output.

In addition, RACF lists the following information from the RACF segment of the user profile for each group to which the user is connected:

- The group name
- The user’s authority in the group
- The user ID of the person who connected the user to this group
- The date the user was connected to this group
- The number of times the user has entered the system with this group as the current connect group
- The default universal access authority
- The date and time the user last entered the system using this group as the current connect group
- The connect attributes (group-related user attributes).
LISTUSER

Details about listing a user's envelope information:

- Listing information about password envelopes:
  - Information about a user's password envelope is displayed only if the user does not have the PROTECTED attribute.
  - If the user’s password is enveloped (regardless of whether password enveloping is enabled), the PASSWORD ENVELOPED=YES line is displayed.
  - If the user’s password is not enveloped and password enveloping is enabled, the PASSWORD ENVELOPED=NO line is displayed.
  - If the user’s password is not enveloped and password enveloping is not enabled, no output line about password enveloping is displayed.

- Listing information about password phrase envelopes:
  - Information about a user's password phrase envelope is displayed only if the user does not have the PROTECTED attribute.
  - If the user's password phrase is enveloped (regardless of whether password phrase enveloping is enabled), the PHRASE ENVELOPED=YES line is displayed.
  - If the user's password phrase is not enveloped and password phrase enveloping is enabled, the PHRASE ENVELOPED=NO line is displayed.
  - If the user's password phrase is not enveloped and password phrase enveloping is not enabled, or if the user has no password phrase, no output line about password phrase enveloping is displayed.

RACF date handling: RACF interprets dates with 2-digit years as follows. (The yy value represents the 2-digit year.)

- If 70 < yy <= 99, the date is interpreted as 19yy.
- If 00 <= yy <= 70, the date is interpreted as 20yy.

Issuing options

The following table identifies the eligible options for issuing the LISTUSER command:

<table>
<thead>
<tr>
<th>As a RACF TSO command?</th>
<th>As a RACF operator command?</th>
<th>With command direction?</th>
<th>With automatic command direction?</th>
<th>From the RACF parameter library?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>

For information on issuing this command as a RACF TSO command, refer to Chapter 3, “RACF TSO commands,” on page 15.

For information on issuing this command as a RACF operator command, refer to Chapter 4, “RACF operator commands,” on page 21.

You must be logged on to the console to issue this command as a RACF operator command.

Related commands

- To list a group profile, see LISTGRP (List group profile)
- To list a data set profile, see LISTDSD (List data set profile)
- To list a general resource profile, see RLIST (List general resource profile) (General resources include terminals, and other resources defined in the class descriptor table.)
- To list information about user ID associations, see RAACLINK (Administer user ID associations).
LISTUSER

- To obtain a list of user profiles, see SEARCH (Search RACF database).

Authorization required

When issuing the LISTUSER command as a RACF operator command, you might require sufficient authority to the proper resource in the OPERCMDS class. See z/OS Security Server RACF Security Administrator's Guide for further information.

To specify the AT keyword, you must have READ authority to the DIRECT.node resource in the RRSFDATA class and a user ID association must be established between the specified node.userid pair(s).

To specify the ONLYAT keyword you must have the SPECIAL attribute, the userid specified on the ONLYAT keyword must have the SPECIAL attribute, and a user ID association must be established between the specified node.userid pair(s) if the user IDs are not identical.

Listing the RACF segment of a user profile: You can always list the details of the RACF segment of your own user profile. To list details of the RACF segment of another user’s profile, one of the following conditions must be true:

- You are the owner of the user’s profile.
- You have the SPECIAL attribute.
- The user’s profile is within the scope of a group in which you have the group-SPECIAL attribute.
- You have the AUDITOR attribute.
- You have READ access to the IRR.LISTUSER resource in the FACILITY class and the user does not have the SPECIAL, AUDITOR, or OPERATIONS attribute.
- You have READ access to an appropriate resource (IRR.LU OWNER.owner or IRR.LU TREE.owner) in the FACILITY class, and both of the following conditions are also true:
  - The user does not have the SPECIAL, AUDITOR, or OPERATIONS attribute. (You can list a PROTECTED user.)
  - You are not excluded from listing the user by the IRR.LU.EXCLUDE.excluded-user resource in the FACILITY class.

For more information about the IRR.LU profiles, see z/OS Security Server RACF Security Administrator's Guide.

To list details of the RACF segment of all RACF-defined user profiles (by specifying the asterisk (*) operand), one of the following conditions must be true for each listed profile:

- You are the owner of the user’s profile. RACF lists the RACF segment for all the user profiles that you own.
- You have the SPECIAL attribute. RACF lists the RACF segment for all user profiles.
- The user’s profile is within the scope of a group in which you have the group-SPECIAL attribute. RACF lists the RACF segment for all the user profiles within the scope of your group.
- You have the AUDITOR attribute. RACF lists the RACF segment for all user profiles.
LISTUSER

- The user's profile is within the scope of a group in which you have the group-AUDITOR attribute. RACF lists the RACF segment for all the user profiles within the scope of your group.
- You have READ access to the IRR.LISTUSER resource in the FACILITY class and the user does not have any of the SPECIAL, AUDITOR or OPERATIONS attributes.

If you have the group-SPECIAL or group-AUDITOR attribute and your installation has assigned security levels and security categories to user profiles, you must have the following to be able to display the RACF segment of a user's profile:
- A security level equal to, or greater than, that in the user profile you are trying to display
- All security categories contained in the user profile you are trying to display contained in your own user profile.

If you have the AUDITOR attribute, or the profile is within the scope of a group in which you have the group-AUDITOR attribute, RACF also lists the value of the UAUDIT/NOUAUDIT operand.

Listing the other segments of a user profile: To list information from segments other than the RACF segment for a user profile, including your own, one of the following conditions must be true:
- You have the SPECIAL or AUDITOR attribute
- You have at least READ authority to the desired field within the segment through field-level access checking.

Syntax

For the key to the symbols used in the command syntax diagrams, see "Syntax of RACF commands and operands" on page 9. The complete syntax of the LISTUSER command is:

```
[(subsystem-prefix][LISTUSER | LU]
   [ (userid ...) | * ]
   [ AT([node].userid ...) | ONLYAT([node].userid ...) ]
   [ CICS ]
   [ CSDATA ]
   [ DCE ]
   [ DFP ]
   [ EIM ]
   [ KERB ]
   [ LANGUAGE ]
   [ LNOTES ]
   [ NDS ]
   [ NETVIEW ]
   [ NORACF ]
   [ OMVS ]
   [ OPERPARM ]
   [ OVM ]
   [ PROXY ]
   [ TSO ]
   [ WORKATTR ]
```
LISTUSER

For information on issuing this command as a RACF TSO command, refer to Chapter 3, “RACF TSO commands,” on page 15.

For information on issuing this command as a RACF operator command, refer to Chapter 4, “RACF operator commands,” on page 21.

Parameters

**subsystem-prefix**

Specifies that the RACF subsystem is the processing environment of the command. The subsystem prefix can be either the installation-defined prefix for RACF (1–8 characters) or, if no prefix has been defined, the RACF subsystem name followed by a blank. If the command prefix was registered with CPF, you can use the MVS command D OPDATA to display it or you can contact your RACF security administrator.

Only specify the subsystem prefix when issuing this command as a RACF operator command. The subsystem prefix is required when issuing RACF operator commands.

**userid** | *

Specifies the user ID of one or more RACF-defined users. If you specify more than one user ID, you must enclose the list of user IDs in parentheses.

* Specifies that you want to list information contained in all RACF-defined user profiles to which you have the required authority.

**Important:** On a system with many users defined, the use of * may result in a large amount of output and may not be useful to a user issuing the command. It may be more appropriate for the user to browse the output of IRRDBU00 (database unload) or to write a program to process the IRRDBU00 output and produce a report showing only the subset of information that is of interest to the user. The processing of output of LISTUSER by programs is not supported nor recommended by IBM. If you want a listing of all the groups for use by a program you should instead have the program process the output from IRRDBU00, RACROUTE REQUEST=EXTRACT, or ICHEINTY.

The userid value or an asterisk (*) must be specified if you specify any other operand in the LISTUSER command, and must be the first operand following LISTUSER.

If you enter LISTUSER and specify one or more user IDs, or an asterisk (*), without specifying an additional operand, RACF lists only the RACF segment information from the specified profiles.

If you enter only LISTUSER, RACF lists only the RACF segment information from your own user profile.

**Note:** You cannot use the LISTUSER command for user IDs that have mixed-case characters, such as irrcerta, irrsitec, and irrmulti (which are associated with digital certificates).

**AT | ONLYAT**

The AT and ONLYAT keywords are only valid when the command is issued as a RACF TSO command.
LISTUSER

AT([node].userid ...)
Specifies that the command is to be directed to the node specified by node, where it runs under the authority of the user specified by userid in the RACF subsystem address space.

If node is not specified, the command is directed to the local node.

ONLYAT([node].userid ...)
LISTUSER is not eligible for automatic command direction. If you specify the ONLYAT keyword, the effect is the same as if you specified the AT keyword.

CICS
Specifies that you want to list the information contained in the CICS segment of the user's profile.

The details RACF lists from the CICS segment of the user's profile are:
- The classes assigned to this operator to which BMS messages are sent.
- Whether the operator are forced off when an XRFSOFF takeover occurs.
- The operator identification.
- The priority of the operator.
- The time in hours and minutes that the operator is allowed to be idle before being signed off.
- Resource security level (RSL) keys, if any are assigned to the user. If 99 is displayed, this indicates that all RSL keys are assigned to the user (1–24, inclusive). If 0 is displayed, no RSL keys are assigned to the user.
- Transaction security level (TSL) keys, if any are assigned to the user. If 99 is displayed, this indicates that all TSL keys are assigned to the user (1–64, inclusive). If 0 is displayed, no TSL keys are assigned to the user.

CSDATA
Specifies that you want to list custom field information for this user. The custom field information in the CSDATA segment for this user was added using the ADDUSER and ALTUSER commands.

Usage for each custom field is defined using the CFDEF operand of the RDEFINE command for resource profiles in the CFIELD class. Contact your security administrator to see how custom fields are used at your installation. For more information about custom fields, see z/OS Security Server RACF Security Administrator's Guide

DCE
Specifies that you want to list the information contained in the DCE segment of the user's profile.

The details that RACF lists from the DCE segment are:
- The DCE universal unique identifier
- The DCE principal name
- The DCE home cell name
- The DCE home cell universal unique identifier
- The DCE AUTOLOGIN indicator.

If there is no DCENAME or HOMECELL for this segment, the field name is not displayed. However, if UUID or HOMEUUID was not specified when the DCE segment was added to the user profile, the word NONE appears in the listing.
LISTUSER

DFP
Specifies that you want to list the information contained in the DFP segment of
the user’s profile.

The details RACF lists from the DFP segment of the user’s profile are:
• The user's default data class
• The user's default management class
• The user's default storage class
• The data management data application for the user.

EIM
Specifies that the Enterprise Identity Mapping (EIM) segment information should
be listed.

KERB
Specifies that you want to list the information contained in the KERB segment
of the user’s profile.

The details that RACF lists from the KERB segment of the user’s profile are:
• The encryption value settings (ENCRYPT values or NOENCRYPT)
• The local kerberos-principal-name (KERBNAME)
• The max-ticket-life associated with this local principal (MAXTKTLFE)
• The current z/OS Network Authentication Service key version (KEY VERSION)
• The authenticator used to generate the current user’s z/OS Network
Authentication Service keys (KEY FROM)
  – When PASSWORD is displayed, the current keys were derived from the
    user’s password.
  – When PHRASE is displayed, the current keys were derived from the user's
    password phrase.

LANGUAGE
Specifies that you want to list the information contained in the LANGUAGE
segment of the user’s profile.

The 3-character language code and, if defined, the 24-character language
name, is displayed. NOT SPECIFIED indicates that no language has been
specified.

If the code is displayed without a name, one of the following is true:
• The MVS message service was not active
• The language was not active.

If the language code equals the language name, one of the following is true:
• There was no language name defined on your system
• The language name was defined to be the same as the language code.

The details RACF lists from the LANGUAGE segment of the user’s profile are:
• The user’s primary language, if one has been specified
• The user’s secondary language, if one has been specified.

LNOTES
Specifies that you want to list the information for the Lotus Notes for z/OS
short-name, which is contained in the LNOTES segment of the user’s profile.

NDS
Specifies that you want to list the information for the Novell Directory Services
for OS/390 user-name, which is contained in the NDS segment of the user’s
profile.
LISTUSER

NETVIEW
Specifies that you want to list the information contained in the NETVIEW segment of the user's profile.

The details RACF lists from the NETVIEW segment of the user's profile are:

- The command or command line to be processed by NetView for this operator
- The default MCS console identifier
- Whether security checking should be done for this NetView operator
- Whether this operator can receive unsolicited messages
- The count of operator class values
- The list of NetView scope classes for which this operator has authority
- The number of domains this NetView operator controls
- The list of identifiers of NetView programs in another NetView domain for which this operator has authority
- Whether this operator has administrator authority to the NetView Graphic Monitor Facility (NGMF).

If there is no information in the fields of the NETVIEW segment, the field name is not displayed (with the exception of SIZE, MAXSIZE, and USERDATA).

NORACF
Specifies that you want to suppress the listing of RACF segment information from the user's profile.

If you specify NORACF, you must also specify at least one segment name.

The information displayed as a result of using the NORACF operand is dependent on other operands used in the command. For example, if you use NORACF with TSO or DFP also specified, only that information (TSO or DFP) is displayed. User profiles that do not have at least one of the specified segments appear in the command output.

The information displayed as a result of using the NORACF operand is dependent on other operands used in the command. For example, if you use NORACF with TSO or DFP also specified, only that information (TSO or DFP) is displayed.

When you specify an asterisk (*) in place of the user ID, only user profiles with at least one of the specified segments appear in the command output. (See the description of the userid parameter on page 241 for an important note about specifying an asterisk with the LISTUSER command.)

If you do not specify NORACF, RACF displays the information in the RACF segment of a user profile.

OMVS
Specifies that you want to list the information contained in the OMVS segment of the user's profile.

The details RACF lists from the OMVS segment are:

- The user identifier
- The initial directory pathname
- The program pathname
- The CPU time, in seconds, the user's processes can use
- The address space region size, in bytes, the user's processes can use
- The maximum number of active or open files the user can have
- The maximum number of active processes the user can have
- The maximum number of threads the user can have
- The maximum amount of space, in pages, the user can map in storage.
Note: If CPUTIMEMAX, ASSIZEMAX, FILEPROCMA, PROCUSERMA, THREADSMAX, or MMAPAREAMAX is not specified, or is removed with the ALTUSER command, the word NONE appears in the listing. In such situations, z/OS UNIX uses its system level values for limit values.

If there is no HOME or PROGRAM information, the field name is not displayed. However, the word NONE appears in the listing if the UID was not specified, or if the UID was removed using the NOUID operand on the ALTUSER command.

OPERPARM
Specifies that you want to list the information contained in the OPERPARM segment of the user’s profile.

The details RACF lists from the OPERPARM segment of the user’s profile are:
- The alternate console group (ALTGRP)
- The operator authority (AUTH)
- Whether the console receives messages that can be automated in a sysplex environment.
- The system name for commands from this console (CMDSYS)
- Whether, and what kind of, delete operator messages are received (DOM)
- The searching key (KEY)
- The message level information (LEVEL)
- Whether system command responses are logged (LOGCMDRESP)
- The message format (MFORM)
- Whether this console is assigned a migration ID (MIGID)
- Event information (MONITOR)
- The systems this console can receive undirected messages from (MSCOPE)
- Routing code information (ROUTCODE)
- Storage information (STORAGE)
- Whether this console receives undeliverable messages (UD).

If there is no information in a field in the user’s profile for this segment, the field name is not displayed. However, if no value was specified for STORAGE when the OPERPARM segment was added to the user profile, STORAGE=0 appears in the listing.

OVM
Specifies that you want to list the information contained in the OVM segment of the user’s profile.

The details that RACF lists from the OVM segment are the z/OS UNIX System Services user’s:
- User identifier
- Initial directory pathname
- Program pathname
- File system root name.

If there is no HOME, PROGRAM, or FSROOT information, the field name is not displayed. However, the word NONE appears in the listing if the UID was not specified, or if the UID was removed using the NOUID operand on the ALTUSER command.

PROXY
Specifies that PROXY segment information should be listed.

The BINDPW password value will not be listed. If a BINDPW password value is defined for a user, LISTUSER will display YES for the PROXY segment BINDPW attribute. If no BINDPW password value has been defined, LISTUSER will display NO for the PROXY segment BINDPW attribute.
LISTUSER

TSO
Specifies that you want to list the information contained in the TSO segment of the user's profile.

The details RACF lists from the TSO segment of the user's profile are:
- The user's default account number when logging on from the TSO/E logon panel
- The destination ID for SYSOUT data sets
- The user's default HOLDCLASS
- The user's default JOBCLASS
- The user's default MSGCLASS
- The user's default SYS
- The maximum region size
- The default region size
- The logon procedure name
- The unit name
- The optional user data
- The user's security label
- The default command to be run during the TSO/E logon.

If there is no information in the fields of the TSO segment, the field name is not displayed (with the exception of SIZE, MAXSIZE, and USERDATA).

WORKATTR
Specifies that you want to list the information contained in the WORKATTR segment of the user’s profile.

The details RACF lists for the distribution information from the user’s WORKATTR segment are:
- The name of the user (WANAME)
- The building (WABLDG)
- The department (WADEPT)
- The room (WAROOM)
- Up to four additional lines of output distribution information (WAADDRn)
- An account number for APPC/MVS processing (WAACCNT).

Examples

Table 17. LISTUSER examples

<table>
<thead>
<tr>
<th>Example 1</th>
<th>Operation</th>
<th>User DAF0 wants to list the user attributes from the RACF segment of her user profile.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Known</td>
<td>User DAF0 is RACF-defined. User DAF0 wants to issue the command as a RACF TSO command.</td>
<td></td>
</tr>
<tr>
<td>Command</td>
<td>LISTUSER</td>
<td></td>
</tr>
<tr>
<td>Defaults</td>
<td>DAF0 (userid)</td>
<td></td>
</tr>
<tr>
<td>Output</td>
<td>See Figure 19</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Example 2</th>
<th>Operation</th>
<th>User CALTMANN wants to list the user attributes from the RACF segment of profiles for users IBMUSER, CALTMANN, and DAF0.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Known</td>
<td>User CALTMANN has the SPECIAL and AUDITOR attributes. User CALTMANN wants to issue the command as a RACF TSO command.</td>
<td></td>
</tr>
<tr>
<td>Command</td>
<td>LISTUSER (IBMUSER CALTMANN DAF0)</td>
<td></td>
</tr>
<tr>
<td>Defaults</td>
<td>None.</td>
<td></td>
</tr>
<tr>
<td>Output</td>
<td>See Figure 20</td>
<td></td>
</tr>
</tbody>
</table>
Table 17. LISTUSER examples (continued)

Example 3

Operation User ADM1 wants to list the user attributes from the RACF segment and TSO segment of the profile for user DAF0.

Known User ADM1 has the SPECIAL and AUDITOR attributes.

User DAF0 is defined to RACF with authority to use TSO.

User ADM1 wants to issue the command as a RACF TSO command.

Command LISTUSER DAF0 TSO

Defaults None.

Output See Figure 21.

Example 4

Operation User ADM1 wants to list the user attributes from only the TSO segment of the profile for user DAF0.

Known User ADM1 has the SPECIAL and AUDITOR attributes.

User DAF0 is defined to RACF with authority to use TSO.

User ADM1 wants to issue the command as a RACF TSO command.

Command LISTUSER DAF0 NORACF TSO

Defaults None.

Output See Figure 22.

Example 5

Operation User ADM1 wants to list the user attributes from the RACF segment and DFP segment of the profile for user DAF0.

Known User ADM1 has the SPECIAL and AUDITOR attributes.

User DAF0 is defined to RACF and DAF0's profile contains a DFP segment.

User ADM1 wants to issue the command as a RACF TSO command.

Command LISTUSER DAF0 DFP

Defaults None.

Output See Figure 23.

Example 6

Operation User ADM1 wants to list the user attributes from only the DFP segment of the profile for user DAF0.

Known User ADM1 has the SPECIAL and AUDITOR attributes.

User DAF0 is defined to RACF and DAF0's profile contains a DFP segment.

User ADM1 wants to issue the command as a RACF TSO command.

Command LISTUSER DAF0 NORACF DFP

Defaults None.

Output See Figure 24.

Example 7

Operation User ADM1 wants to list the user attributes from only the CICS segment of the profile for user DAF0.

Known User DAF0 is defined to RACF and DAF0's profile contains a CICS segment.

User running CICS in a distributed environment.

User ADM1 wants to issue the command as a RACF TSO command.

Command LISTUSER DAF0 NORACF CICS

Defaults None.

Output See Figure 25.
LISTUSER

Table 17. LISTUSER examples (continued)

Example 8  
Operation User ADM1 wants to list the user attributes from only the LANGUAGE segment of the profile for user DAF0.
Known User ADM1 has the SPECIAL and AUDITOR attributes.

User DAF0 is defined to RACF and DAF0’s profile has American English (language code ENU) defined as her primary language and German (language code DEU) defined as her secondary language.

User ADM1 wants to issue the command as a RACF TSO command.

Command LISTUSER DAF0 NORACF LANGUAGE
Defaults None.
Output See Figure 26.

Example 9  
Operation User ADM1 wants to list the user attributes from only the OPERPARM segment of the profile for user DAF0.
Known User ADM1 has the SPECIAL and AUDITOR attributes.

User DAF0 is defined to RACF and DAF0’s profile contains an OPERPARM segment.

User ADM1 wants to issue the command as a RACF TSO command.

Command LISTUSER DAF0 NORACF OPERPARM
Defaults None.
Output See Figure 27.

Example 10  
Operation User ADM1 wants to list the user attributes from the OMVS segment of the profile for user CSMITH.
Known User ADM1 has the SPECIAL attribute.

User CSMITH is defined to RACF and CSMITH’s profile contains an OMVS segment.

User ADM1 wants to issue the command as a RACF TSO command.

Command LISTUSER CSMITH OMVS NORACF
Defaults None.
Output See Figure 28.

Example 11  
Operation User ADM1 wants to list the user attributes from the OMVS segment of the profile for user CSMITH.
Known User ADM1 has the SPECIAL attribute.

User CSMITH is defined to RACF and CSMITH’s profile contains an OMVS segment, but there was no value specified for HOME or PROGRAM in the OMVS segment for this profile. Defaults were used.

User ADM1 wants to issue the command as a RACF TSO command.
Note: If the user also has no user limits because the defaults were taken, CPUTIMEMAX, ASSIZEMAX, FILEPROCMAX, PROCUSERMAX, THREADSMAX, and MMAPAREAMAX will display NONE as their value.

Command LISTUSER CSMITH OMVS NORACF
Defaults None.
Output See Figure 29.

Example 12  
Operation User ADM1 wants to list the DCE segment for user CSMITH.
Known User ADM1 has the SPECIAL attribute.

Command LISTUSER CSMITH NORACF DCE
Defaults None.
Output See Figure 30.
Table 17. LISTUSER examples (continued)

Example 13  
Operation A security administrator lists the KERB segment of the altered RACF user profile for RONTOMS.
Known The administrator wants to list the information contained in the KERB segment of the altered RACF user profile.
Command LISTUSER RONTOMS NORACF KERB
Defaults None.
Output See Figure 31.

Example 14  
Operation A security administrator lists the PROXY segment of the altered RACF user profile for MRSERVER.
Known The administrator wants to list the information contained in the PROXY segment of the altered RACF user profile.
Command LISTUSER MRSERVER PROXY NORACF
Defaults None.
Output See Figure 32.

Example 15  
Operation A security administrator lists the EIM segment of the RACF user profile for KCROVE.
Known User ADM1 has the SPECIAL attribute.
Command LISTUSER KCROVE EIM NORACF
Defaults None.
Output See Figure 33.

Example 16  
Operation User ADM1 wants to list the status of the RACF user profile for UPWENV who has an enveloped password and an enveloped password phrase.
Known User ADM1 has the SPECIAL attribute. User UPWENV does not have the PROTECTED attribute.
Command LISTUSER UPWENV
Defaults None.
Output See Figure 34.

Example 17  
Operation User SECADM wants to list the custom field information for user ANDREW.
Known User SECADM has the SPECIAL attribute.
Command LISTUSER ANDREW CSDATA NORACF
Output See Figure 35.
LISTUSER

LISTUSER
USER=DAF0  NAME=D.M.BROWN  OWNER=IBMUSER  CREATED=05.228
DEFAULT-GROUP=RESEARCH  PASSDATE=05.228  PASS-INTERVAL=30  PHRASEDATE=05.231
PASSWORD ENVELOPED=NO
ATTRIBUTES=ADSP
ATTRIBUTES=PASSPHRASE
REVOKE DATE=NONE  RESUME DATE=NONE
LAST-ACCESS=05.228/13:31:11
CLASS AUTHORIZATIONS=NONE
NO-INSTALLATION-DATA
NO-MODEL-NAME
LOGON ALLOWED (DAYS) (TIME)
--------------------------------------------
ANYDAY  ANYTIME
GROUP=RESEARCH  AUTH=JOIN  CONNECT-OWNER=IBMUSER  CONNECT-DATE=05.228
  CONNECTS=01  UACC=READ  LAST-CONNECT=05.228/13:31:11
  CONNECT ATTRIBUTES=NONE
  REVOKE DATE=NONE  RESUME DATE=NONE
GROUP=PAYROLLB  AUTH=CREATE  CONNECT-OWNER=IBMUSER  CONNECT-DATE=05.228
  CONNECTS=00  UACC=READ  LAST-CONNECT=UNKNOWN
  CONNECT ATTRIBUTES=NONE
  REVOKE DATE=NONE  RESUME DATE=NONE
SECURITY-LEVEL=NONE SPECIFIED
CATEGORY-AUTHORIZATION
NONE SPECIFIED

Figure 19. Example 1: Output for LISTUSER

LISTUSER (IBMUSER CALTMANN DAF0)
USER=IBMUSER  NAME=G. SMITH  OWNER=IBMUSER  CREATED=05.163
DEFAULT-GROUP=SYS1  PASSDATE=05.220  PASS-INTERVAL=N/A  PHRASEDATE=05.231
PASSWORD ENVELOPED=NO
ATTRIBUTES=SPECIAL OPERATIONS
ATTRIBUTES=PASSPHRASE AUDITOR
REVOKE DATE=NONE  RESUME DATE=NONE
LAST-ACCESS=05.146/15:45:23
CLASS AUTHORIZATIONS=NONE
NO-INSTALLATION-DATA
NO-MODEL-NAME
LOGON ALLOWED (DAYS) (TIME)
--------------------------------------------
ANYDAY  ANYTIME
GROUP(SYS1  AUTH=JOIN  CONNECT-OWNER=IBMUSER  CONNECT-DATE=04.263
  CONNECTS=456  UACC=READ  LAST-CONNECT=05.146/15:45:23
  CONNECT ATTRIBUTES=NONE
  REVOKE DATE=NONE  RESUME DATE=NONE
SECURITY-LEVEL=NONE SPECIFIED
CATEGORY-AUTHORIZATION
NONE SPECIFIED
SECURITY-LABEL=NONE SPECIFIED

Figure 20. Example 2: Output for LISTUSER (IBMUSER CALTMANN DAF0) (Part 1 of 2)
### Figure 20. Example 2: Output for LISTUSER (IBMUSER CALTMAN DAF0) (Part 2 of 2)

<table>
<thead>
<tr>
<th>USER</th>
<th>NAME</th>
<th>OWNER</th>
<th>CREATED</th>
</tr>
</thead>
<tbody>
<tr>
<td>CALTMANN</td>
<td>C. ALTMANN</td>
<td>IBMUSER</td>
<td>05.144</td>
</tr>
<tr>
<td>DEFAULT-GROUP</td>
<td>RESEARCH</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PASSDATE</td>
<td>00.000</td>
<td>PASS-INTERVAL</td>
<td>254</td>
</tr>
<tr>
<td>PHRASEDATE</td>
<td>05.231</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PASSWORD</td>
<td>ENVELOPED=NO</td>
<td>ATTRIBUTES</td>
<td>PASSPHRASE AUDITOR</td>
</tr>
<tr>
<td>REVOKE DATE</td>
<td>RESUME DATE=NONE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LAST-ACCESS</td>
<td>05.146/16:16:14</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CLASS AUTHORIZATIONS</td>
<td>USER</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NO-INSTALLATION-DATA</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MODEL-NAME</td>
<td>ALLENA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LOGON ALLOWED</td>
<td>ANYDAY ANYTIME</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GROUP</td>
<td>RESEARCH</td>
<td>AUTH=JOIN</td>
<td>CONNECT-OWNER=IBMUSER CONNECT-DATE=05.144</td>
</tr>
<tr>
<td>CONNECTS</td>
<td>01 UACC=READ</td>
<td>LAST-CONNECT=05.146/16:16:14</td>
<td></td>
</tr>
<tr>
<td>CONNECT ATTRIBUTES</td>
<td>OPERATIONS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>REVOKE DATE</td>
<td>RESUME DATE=NONE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SECURITY-LEVEL</td>
<td>NONE SPECIFIED</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CATEGORY-AUTHORIZATION</td>
<td>NONE SPECIFIED</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SECURITY-LABEL</td>
<td>NONE SPECIFIED</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>USER</th>
<th>NAME</th>
<th>OWNER</th>
<th>CREATED</th>
</tr>
</thead>
<tbody>
<tr>
<td>DAF0</td>
<td>D.M. BROWN</td>
<td>IBMUSER</td>
<td>05.144</td>
</tr>
<tr>
<td>DEFAULT-GROUP</td>
<td>RESEARCH</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PASSDATE</td>
<td>00.000</td>
<td>PASS-INTERVAL</td>
<td>254</td>
</tr>
<tr>
<td>PHRASEDATE</td>
<td>05.231</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PASSWORD</td>
<td>ENVELOPED=NO</td>
<td>ATTRIBUTES</td>
<td>ADSP</td>
</tr>
<tr>
<td>REVOKE DATE</td>
<td>RESUME DATE=NONE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LAST-ACCESS</td>
<td>05.146/15:11:31</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CLASS AUTHORIZATIONS</td>
<td>NONE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NO-INSTALLATION-DATA</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MODEL-NAME</td>
<td>ALLENA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LOGON ALLOWED</td>
<td>ANYDAY ANYTIME</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Chapter 5. RACF command syntax** 251
LISTUSER DAF0 TSO
USER=DAF0 NAME=D.M.BROWN OWNER=IBMUSER CREATED=05.228
DEFAULT-GROUP=RESEARCH PASSDATE=05.231 PASS-INTERVAL=30 PHRASEDATE=05.231
PASSWORD ENVELOPED=NO
ATTRIBUTES=ADSP
ATTRIBUTES=PASSPHRASE
REVOKE DATE=NONE RESUME DATE=NONE
LAST-ACCESS=05.228/13:31:11
CLASS AUTHORIZATIONS=NONE
NO-INSTALLATION-DATA
NO-MODEL-NAME
LOGON ALLOWED (DAYS) (TIME)
-------------------------------------------------------------
ANYDAY ANYTIME
GROUP=RESEARCH AUTH=JOIN CONNECT-OWNER=IBMUSER CONNECT-DATE=05.228
CONNECTS= 01 UACC=READ LAST-CONNECT=05.228/13:31:11
CONNECT ATTRIBUTES=NONE
REVOKE DATE=NONE RESUME DATE=NONE
GROUP=PAYROLLB AUTH=CREATE CONNECT-OWNER=IBMUSER CONNECT-DATE=05.228
CONNECTS= 00 UACC=READ LAST-CONNECT=UNKNOWN
CONNECT ATTRIBUTES=NONE
REVOKE DATE=NONE RESUME DATE=NONE
SECURITY-LEVEL=NONE SPECIFIED
CATEGORY-AUTHORIZATION
NONE SPECIFIED
SECURITY-LABEL=NONE SPECIFIED
TSO INFORMATION
-----------------------------------------------
ACCTNUM= P00F1V
HOLDCLASS= H
JOBCLASS= I
MSGCLASS= A
PROC= V0LOGON
SIZE= 00001024
MAXSIZE= 00002048
SYSOUTCLASS = A
UNIT= SYSDA
USERDATA= 0000

Figure 21. Example 3: Output for LISTUSER DAF0 TSO

LISTUSER DAF0 NORACF TSO
USER=DAF0
TSO INFORMATION
ACCTNUM= P00F1V
HOLDCLASS= H
JOBCLASS= I
MSGCLASS= A
PROC= V0LOGON
SIZE= 00001024
MAXSIZE= 00002048
SYSOUTCLASS = A
UNIT= SYSDA
USERDATA= 0000

Figure 22. Example 4: Output for LISTUSER NORACF TSO
LISTUSER DAF0 DFP
USER=DAF0
NAME=D.M.BROWN
OWNER=IBMUSER
CREATED=05.228
DEFAULT-GROUP=RESEARCH
PASSDATE=05.228
PASS-INTERVAL=30
PHRASEDATE=05.231
PASSWORD ENVELOPED=NO
ATTRIBUTES=ADSP
ATTRIBUTES=PASSPHRASE
REVOKE DATE=NONE
RESUME DATE=NONE
LAST-ACCESS=05.228/13:31:11
CLASS AUTHORIZATIONS=NONE
NO-INSTALLATION-DATA
NO-MODEL-NAME
LOGON ALLOWED (DAYS) (TIME)
--------------------------------------------
ANYDAY
GROUP=RESEARCH
AUTH=JOIN
CONNECT-OWNER=IBMUSER
CONNECT-DATE=05.228
CONNECTS= 01
UACC=READ
LAST-CONNECT=05.228/13:31:11
CONNECT ATTRIBUTES=NONE
REVOKE DATE=NONE
RESUME DATE=NONE
GROUP=PAYROLLB
AUTH=CREATE
CONNECT-OWNER=IBMUSER
CONNECT-DATE=05.228
CONNECTS= 00
UACC=READ
LAST-CONNECT=UNKNOWN
CONNECT ATTRIBUTES=NONE
REVOKE DATE=NONE
RESUME DATE=NONE
SECURITY-LEVEL=NONE SPECIFIED
CATEGORY-AUTHORIZATION
NONE SPECIFIED
SECURITY-LABEL=NONE SPECIFIED
DFP INFORMATION
-------------------------------------------
MGMTCLAS= DFP5MGMT
STORCLAS= DFP5STOR
DATACLAS= DFP5DATA
DATAAPPL= DFP5APPL

Figure 23. Example 5: Output for LISTUSER DAF0 DFP

LISTUSER DAF0 NORACF DFP
USER=DAF0
DFP INFORMATION
-------------------------------------------
MGMTCLAS= DFP5MGMT
STORCLAS= DFP5STOR
DATACLAS= DFP5DATA
DATAAPPL= DFP5APPL

Figure 24. Example 6: Output for LISTUSER DAF0 NORACF DFP
LISTUSER DAF0 NORACF CICS
USER=TEST
CICS INFORMATION
----------------
OPCLASS=001
OPIDENT= ID2
OPPRTY= 00010
TIMEOUT= 02:30 (HH:MM)
XRFSOFF= NOFORCE
RSLKEYS= 00001 00003 00005 00007 00009 00011 00002
00018 00016 00014 00012 00023 00021 00019
00017 00015 00013
TSLKEYS= 00001 00003 00005 00007 00009 00011 00002
00004 00006 00008 00010 00024 00022 00020
00018 00016 00014 00012 00023 00021 00019
00038 00035 00036 00032 00064 00041 00063
00043 00048 00051 00042 00055 00062 00044
00061 00060 00059 00058

Figure 25. Example 7: Output for LISTUSER DAF0 NORACF CICS

LISTUSER DAF0 NORACF LANGUAGE
USER=DAF0
LANGUAGE INFORMATION
----------------------
PRIMARY LANGUAGE: ENU
SECONDARY LANGUAGE: DEU
READY

Figure 26. Example 8: Output for LISTUSER DAF0 NORACF LANGUAGE

LU DAF0 NORACF OPERPARM
USER=DAF0
OPERPARM INFORMATION
---------------------
STORAGE= 00002
AUTH= IO
ROUTCODE= ALL
LEVEL= ALL
MFORM= T J M
MONITOR= JOBNAMEST SESST
MIGID= YES
DOM= NORMAL
KEY= MCS2
CMDSYS= SYS1
MSCOPE= *ALL
UD= YES
HC= YES
INTIDS= YES
UNKNIDS= YES
READY

Note: With the exception of the STORAGE operand, if a field has no value in the OPERPARM segment, no value appears for the field in the listing. If there is an OPERPARM segment and the storage is not specified, 00000 appears in the listing. When an extended MCS console session is established, the values for STORAGE is 1.

Figure 27. Example 9: Output for LISTUSER DAF0 NORACF OPERPARM
LISTUSER CSMITH OMVS NORACF
USER = CSMITH
OMVS INFORMATION
----------------
UID= 0000000024
HOME= /u/CSMITH
PROGRAM= /u/CSMITH/bin/myshell
CPUTIMEMAX= 0010000000
ASSIZEMAX= NONE
FILEPROCMAx= 0000050000
PROCUSERMAX= NONE
THREADSMAX= NONE
MMAPAREAMAX= 0016777216

Figure 28. Example 10: Output for listing OMVS user information

LISTUSER CSMITH OMVS NORACF
USER=CSMITH
OMVS INFORMATION
----------------
UID= 0000000024
CPUTIMEMAX= NONE
ASSIZEMAX= NONE
FILEPROCMAX= NONE
PROCUSERMAX= NONE
THREADSMAX= NONE
MMAPAREAMAX= NONE

Figure 29. Example 11: Output for LISTUSER CSMITH OMVS NORACF (Using Defaults)

LISTUSER CSMITH NORACF DCE
USER=CSMITH
DCE INFORMATION
---------------
UUID= 004386ea-ebb6-lec3-bcae-10005ac90feb
DCENAME= charlie
HOME CELL UUID= 003456aab-ecb7-7de3-ebda-95531ed63dae
HOME CELL= /.../hootie.scarol.ibm.com
DCE AUTOLOGIN= NO

Figure 30. Example 12: Output for LISTUSER CSMITH NORACF DCE

LISTUSER RONTOMS NORACF KERB
USER=RONTOMS
KERB INFORMATION
----------------
KERBNAME= KerberizedUser
KEY FROM= PASSWORD
KEY VERSION= 001
KEY ENCRYPTION TYPE= DES DES3 DESD AES128 AES256

Figure 31. Example 13: Output for LISTUSER RONTOMS NORACF KERB
LISTUSER

```
LISTUSER MRSERVER PROXY NORACF
USER=MRSERVER
PROXY INFORMATION
-----------------
LDAPHOST= LDAP://SOME.LDAP.HOST:389
BINDDN= cn=Joe User,ou=Poughkeepsie,o=IBM,c=US
BINDPW= YES
```

Figure 32. Example 14: Output for LISTUSER MRSERVER PROXY NORACF

```
LISTUSER KCROVE EIM NORACF
USER=MRSERVER
EIM INFORMATION
---------------
LDAPPROF= EIMDOMAINALOOKUP
```

Figure 33. Example 15: Output for LISTUSER KCROVE EIM NORACF

```
LISTUSER UPWENV
USER=UPWENV NAME=GREGOR OWNER=IBMUSER CREATED=05.161
DEFAULT-GROUP=SYS1 PASSDATE=00.000 PASS-INTERVAL=254 PHRASEDATE=05.231
PASSWORD ENVELOPED=YES
PHRASE ENVELOPED=YES
ATTRIBUTES=PASSPHRASE
```

Figure 34. Example 16: Output for LISTUSER indicating that the user's password and password phrase are each enveloped

```
LISTUSER ANDREW CSDATA NORACF
USER=ANDREW
CSDATA INFORMATION
--------------------------------------
ACTIVE= NO
HOME ADDRESS= 14 Main Street, Anywhere, IL 01234
EMPLOYEE CODE= FC01B2D8
EMPLOYEE SERIAL= 0000256400
HOME PHONE= 555-555-5555
```

Figure 35. Example 17: Output for listing CSDATA user information
PASSWORD or PHRASE

PASSWORD or PHRASE (Specify user password or password phrase)

This command is usually called the PASSWORD command even though the PHRASE command is a supported alias.

Purpose

Use the PASSWORD command to:

- Change your own password or password phrase to a specified value
- Reset another user's password to a known default value
- Change another user's change interval (the number of days that the user's password and password phrase remain valid)
- Specify a password or password phrase that never expires.

When a user's password is changed, RACF makes sure the new password is not the same as the current password. When SETR PASSWORD(HISTORY) is active, RACF also makes sure the new password is not already in the user's password history list. If the new password does not match one of these passwords, the current password is added to the user's password history list, and the new password is activated.

When a user's password phrase is changed, RACF makes sure the new password phrase is not the same as the current password phrase. When SETR PASSWORD(HISTORY) is active, RACF also makes sure the new password phrase is not already in the user's password phrase history list. If the new password phrase does not match one of these password phrases, the new password phrase is added to the user's password phrase history list, and the new password phrase is activated.

If you use the PASSWORD command to change your own password or password phrase and you have user ID associations with password synchronization defined, the password or password phrase is synchronized. However, if you use the PASSWORD command to change another user's password or password phrase, it is not synchronized.

Attention:

- When the PASSWORD command is issued from ISPF, the TSO command buffer (including password or password phrase data) is written to the ISPLOG data set. As a result, you should not issue this command from ISPF or you must control the ISPLOG data set carefully.
- If the PASSWORD command is issued as a RACF operator command, the command and the password or password phrase data is written to the system log. Therefore, use of PASSWORD as a RACF operator command should either be controlled or you should issue the command as a TSO command.

Issuing options

The following table identifies the eligible options for issuing the PASSWORD command:
**PASSWORD or PHRASE**

<table>
<thead>
<tr>
<th>As a RACF TSO command?</th>
<th>As a RACF operator command?</th>
<th>With command direction?</th>
<th>With automatic command direction?</th>
<th>From the RACF parameter library?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

For information on issuing this command as a RACF TSO command, refer to Chapter 3, “RACF TSO commands,” on page 15.

For information on issuing this command as a RACF operator command, refer to Chapter 4, “RACF operator commands,” on page 21.

You must be logged on to the console to issue this command as a RACF operator command.

**Related commands**

- To change the installation change interval, see `SETROPTS (Set RACF options)`.  
- To establish password synchronization between users, see `RACLINK (Administer user ID associations)`.  

**Authorization required**

When issuing the PASSWORD command as a RACF operator command, you might require sufficient authority to the proper resource in the OPERCMDS class. See *z/OS Security Server RACF Security Administrator’s Guide* for further information.

If you are a RACF-defined user and you are required to provide a RACF user password or password phrase when entering the system, you can change your own password, password phrase, or change interval.

To change another user’s change interval, or to set a password and password phrase (if assigned) that never expire, you must have the SPECIAL attribute, or the user’s profile must be within the scope of a group in which you have the group-SPECIAL attribute.

To reset another user’s password to the user’s default value, one of the following conditions must be true:

- You have the SPECIAL attribute.
- The user’s profile is within the scope of a group in which you have the group-SPECIAL attribute.
- You are the owner of the user’s profile.

To specify the AT keyword, you must have READ authority to the `DIRECT.node` resource in the RRSFDATA class and a user ID association must be established between the specified `node.userid` pair(s).

To specify the ONLYAT keyword you must have the SPECIAL attribute, the `userid` specified on the ONLYAT keyword must have the SPECIAL attribute, and a user ID association must be established between the specified `node.userid` pair(s) if the user IDs are not identical.

**Syntax**

For the key to the symbols used in the command syntax diagrams, see “Syntax of RACF commands and operands” on page 9. The complete syntax of the
PASSWORD or PHRASE command is:

```
[subsystem-prefix](PASSWORD | PW | PHRASE)
[ AT([(node),userid ...)] | ONLYAT([(node),userid ...)] ]
[ INTERVAL(change-interval) | NOINTERVAL ]
[ PASSWORD(current-password new-password) ]
[ PHRASE('current-password-phrase' 'new-password-phrase') ]
[ USER(userid ...) ]
```

For information on issuing this command as a RACF TSO command, refer to Chapter 3, “RACF TSO commands,” on page 15.

For information on issuing this command as a RACF operator command, refer to Chapter 4, “RACF operator commands,” on page 21.

**Parameters**

**subsystem-prefix**
Specifies that the RACF subsystem is the processing environment of the command. The subsystem prefix can be either the installation-defined prefix for RACF (1–8 characters) or, if no prefix has been defined, the RACF subsystem name followed by a blank. If the command prefix was registered with CPF, you can use the MVS command D OPDATA to display it or you can contact your RACF security administrator.

Only specify the subsystem prefix when issuing this command as a RACF operator command. The subsystem prefix is required when issuing RACF operator commands.

**AT | ONLYAT**
The AT and ONLYAT keywords are only valid when the command is issued as a RACF TSO command.

**AT([(node),userid ...])**
Specifies that the command is to be directed to the node specified by `node`, where it runs under the authority of the user specified by `userid` in the RACF subsystem address space.

If `node` is not specified, the command is directed to the local node.

**ONLYAT([(node),userid ...])**
Specifies that the command is to be directed only to the node specified by `node` where it runs under the authority of the user specified by `userid` in the RACF subsystem address space.

If `node` is not specified, the command is directed only to the local node.

**INTERVAL | NOINTERVAL**

**INTERVAL(change-interval)**
Specifies the number of days during which a user's password and password phrase (if set) remain valid; the value must be 1–254 days.

The INTERVAL value you specify here cannot exceed the system value (if any) that your installation specified using the INTERVAL operand on the SETROPTS command. (The initial system default after RACF initialization is 30 days.)

The INTERVAL value you specify should not be less than the value (if any) that your installation specified using the MINCHANGE operand on the
PASSWORD or PHRASE

SETROPTS command. If this occurs, the user’s password and password phrase (if set) cannot expire until your installation's minimum interval is reached and the user will not be allowed to change them prior to expiration.

If you specify INTERVAL on the PASSWORD command without a change-interval value, RACF uses the system interval value (if any) that your installation specified or the system default.

To specify INTERVAL with USER, you must have the SPECIAL attribute, or the user profile must be within the scope of a group in which you have the group-SPECIAL attribute.

If you specify the interval incorrectly, RACF ignores this operand.

NOINTERVAL

Specifies that neither a user's password nor password phrase (if set) will expire. To specify NOINTERVAL with USER, you must have the SPECIAL attribute, or the user profile must be within the scope of a group in which you have the group-SPECIAL attribute.

Specifying NOINTERVAL without USER defines your own password and password phrase (if set) to never expire.

You can use INTERVAL at any time to reinstate an expiration interval for a user previously defined with NOINTERVAL.

PASSWORD(current-password new-password)

Specifies your current password and the new one you want. If you enter only the PASSWORD operand, you are prompted so you can enter the current and new passwords in print inhibit mode.

The current and new passwords must have different values. When the installation allows mixed-case passwords, the old and new passwords cannot be the same characters with the case changed. If you specify your current password incorrectly, RACF notifies you and ignores the PASSWORD operand.

You can use the PASSWORD operand to change your own password at any time unless it is within the number of days specified by your installation’s minimum change interval.

RACF ignores this operand when you specify the USER operand.

PHRASE('current-password-phrase' 'new-password-phrase')

Specifies your current password phrase and the new one you want. The new password phrase is a text string of up to 100 characters and must be enclosed in single quotation marks.

When the new-password-phrase exit (ICHPWX11) is present and allows it, the password phrase can be 9–100 characters. When ICHPWX11 is not present, the password phrase must be 14–100 characters. Contact your system programmer to find out if your installation uses the new-password-phrase exit (ICHPWX11) or see z/OS Security Server RACF System Programmer’s Guide for programming details.

Restriction: Because the password phrase value is a quoted string, TSO/E does not support your entering it in print inhibit mode. Therefore, you should take care when entering your new password phrase to ensure it is not observed by others.

The current and new password phrases must have different values. If you specify your current password phrase incorrectly, RACF notifies you and ignores the PHRASE operand.
You can use the PHRASE operand to change your own password phrase at any time unless it is within the number of days specified by your installation's minimum change interval.

The following syntax rules apply to all password phrases. You cannot alter these syntax rules but you can specify additional syntax rules if your installation tailors the new-password-phrase exit (ICHPWX11).

Syntax rules for password phrases:
- Maximum length: 100 characters
- Minimum length:
  - 9 characters, when ICHPWX11 is present and allows the new value
  - 14 characters, when ICHPWX11 is not present
- Must not contain the user ID (as sequential uppercase or sequential lowercase characters)
- Must contain at least 2 alphabetic characters (A–Z, a–z)
- Must contain at least 2 non-alphabetic characters (numerics, punctuation, or special characters)
- Must not contain more than 2 consecutive characters that are identical
- If a single quotation mark is intended to be part of the password phrase, you must use two single quotation marks together for each single quotation mark.

If the new-password-phrase exit (ICHPWX11) is present, it can reject the specified password phrase. RACF rejects password phrases shorter than 14 characters unless ICHPWX11 is present and allows the new value.

If the specified password phrase is accepted, it is made the user’s current password phrase and, when SETROPTS PASSWORD(HISTORY) is in effect, it is added to the user’s password phrase history.

RACF ignores this operand when you specify the USER operand.

USER(userid ...)

Specifies one or more users whose passwords are to be reset. Unless you also specify INTERVAL or NOINTERVAL, each user’s password is set to the user’s respective default group name and the password is set expired.

Notes:
1. To change your own password or password phrase, use the PASSWORD or PHRASE operand, not the USER operand.
2. Specifying USER with your own user ID resets your password to the name of your default group and sets the password as expired, unless you also specify INTERVAL or NOINTERVAL.
3. If you specify USER with the PASSWORD or PHRASE operand, the PASSWORD or PHRASE operand is ignored.

Examples

Table 18. PASSWORD examples

| Example 1 | Operation | User AEH0 wants to change his password from XY262 to YZ344 and increase his change interval to 60 days. |
| Known | User AEH0 is RACF-defined. |
| Command | PASSWORD PASSWORD(XY262 YZ344) INTERVAL(60) |

The maximum installation change-interval is at least 60 days.
Table 18. PASSWORD examples (continued)

Example 2

**Operation**
User ADM1 wants to reset the passwords for users CD0 and DAF0 to the names of their default group.

**Known**
User ADM1 has the group-SPECIAL attribute in group PAYROLL. Group PAYROLL is the owning group of users CD0 and DAF0.

Users CD0 and DAF0 are RACF-defined.

User ADM1 wants to issue the command as a RACF TSO command.

**Command**
PASSWORD USER(CD0 DAF0)

Example 3

**Operation**
User ADM1 wants to set a password that never expires for user CD2. User ADM1 wants to direct the command to run under the authority of CHERYLB at node ALBNY and prohibit the command from being automatically directed to other nodes.

**Known**
Users ADM1 and CHERYLB at ALBNY have the SPECIAL attribute.

User CD2 is RACF-defined on node ALBNY. Users ADM1 and CHERYLB at ALBNY have an already established user ID association.

User ADM1 wants to issue the command as a RACF TSO command.

**Command**
PASSWORD USER(CD2) NOINTERVAL ONLYAT(ALBNY.CHERYLB)

**Results**
The command is only processed at node ALBNY and not automatically directed to any other nodes in the RRSF configuration.

Example 4

**Operation**
Bob wants to change his password from pass1 to word1 on both his user IDs. His user IDs are BOB1 on MVS01 and BOB2 on MVS02.

**Known**
Bob has a peer user ID association with password synchronization established between his two user IDs. Bob wants to issue the command as a RACF TSO command from MVS01.

**Command**
PASSWORD PASSWORD(pass1 word1)

**Results**
The command is processed on MVS01 and the password is changed for user ID BOB1. The password is also changed for user ID BOB2 at MVS02.
PERMIT (Maintain resource access lists)

Purpose

Use the PERMIT command to maintain the lists of users and groups authorized to access a particular resource. RACF provides two types of access lists: standard and conditional.

**Standard Access List:** The standard access list includes the user IDs and group names authorized to access the resource and the level of access granted to each.

**Conditional Access List:** The conditional access list includes the user IDs and group names authorized to access the resource and the level of access granted to each when a certain condition is met. The conditions that can be specified are:

- The name of the program the user must be executing
- The name of the terminal by which the user entered the system
- The name of the JES input device through which the user entered the system
- The name of the system console from which the request was originated
- The name of the APPC partner LU (logical unit) from which the transaction program originated
- The system identifier (SMFID) of the system on which the user is loading the controlled program
- The SERVAUTH profile name that protected the network access security zone name containing the IP address by which the user entered the system
- An application-specific CRITERIA name and value.

RACF considers the conditional access list if one of the following is true:

- The class specified in the condition is active (for the SERVAUTH, TERMINAL, JESINPUT, CONSOLE, or APPCPORT conditions).
- The RACF program control facility is active (for the PROGRAM or the SYSID condition). The RACF program control facility is activated by your installation using SETROPTS WHEN(PROGRAM) command.
- An application-specific CRITERIA name and value is specified on the RACROUTE REQUEST=FASTAUTH request.

If one of the criteria above is met, RACF uses both the standard and conditional access lists when it checks a user’s authority to access a resource; otherwise RACF uses only the standard access list. For more information on conditional access lists or program control, see [z/OS Security Server RACF Security Administrator’s Guide](https://pubs.opengroup.org/onlinepubs/009695399/appendix-g.html).

You can maintain either the standard access list or the conditional access list with a single PERMIT command. Changing both requires you to issue PERMIT twice, with one exception. You can change individual names in one access list and copy the other access list from another profile on one PERMIT command.

Using PERMIT, you can make the following changes to either a standard access list or a conditional access list:

- Give authority to access a discrete or generic resource profile to specific RACF-defined users or groups
- Remove authority to access a discrete or generic resource profile from specific users or groups
PERMIT

- Change the level of access authority to a discrete or generic resource profile for specific users or groups
- Copy the list of authorized users from one discrete or generic resource profile to another profile of either type and modify the new list as you require
- Delete an existing access list.

Using PERMIT to modify an automatic TAPEVOL profile changes the profile to nonautomatic. For more information about TAPEVOL profiles, see z/OS Security Server RACF Security Administrator's Guide

To have changes take effect after updating a user’s access to a generic profile, one of the following steps is required:
- If the command was issued for a data set profile, the user of the data set issues the LISTDSD command:
  
  `LISTDSD DA(data-set-protected-by-the-profile) GENERIC`

  **Note:** Use the data set name, not the profile name.
- The security administrator issues the SETROPTS command:
  
  `SETROPTS GENERIC(class-name) REFRESH`

  See SETROPTS command for authorization requirements.
- The user of the data set or resource logs off and logs on again.

  **Note:** For more information, refer to z/OS Security Server RACF Security Administrator's Guide

### Issuing options

The following table identifies the eligible options for issuing the PERMIT command:

<table>
<thead>
<tr>
<th>As a RACF TSO command?</th>
<th>As a RACF operator command?</th>
<th>With command direction?</th>
<th>With automatic command direction?</th>
<th>From the RACF parameter library?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

For information on issuing this command as a RACF TSO command, refer to Chapter 3, “RACF TSO commands,” on page 15.

For information on issuing this command as a RACF operator command, refer to Chapter 4, “RACF operator commands,” on page 21.

You must be logged on to the console to issue this command as a RACF operator command.

### Related commands

- To specify the UACC for a data set profile, see ADDSD (Add data set profile) (when creating a new profile), or ALTDSD (Alter data set profile) (to change an existing profile).
- To specify the UACC for a general resource (such as a terminal), see RDEFINE (Define general resource profile) (when creating a new profile), or RALTER (Alter general resource profile) (to change an existing profile).
- To obtain a list of profiles, see SEARCH (Search RACF database).
- To list a data set profile, see LISTDSD (List data set profile).
Authorization required

When issuing the PERMIT command as a RACF operator command, you might require sufficient authority to the proper resource in the OPERCMDS class. See z/OS Security Server RACF Security Administrator's Guide for further information.

To perform any of the PERMIT functions, you must have sufficient authority over the resource. RACF makes the following checks until one of the conditions is met:

- You have the SPECIAL attribute.
- The profile is within the scope of a group in which you have the group-SPECIAL attribute.
- You are the owner of the resource.
- If the resource belongs to the DATASET class, the high-level qualifier of the profile name (or the qualifier supplied by the naming conventions routine or a command installation exit) is your user ID.
- If the resource belongs to the DATASET class, you must be the current owner of the profile or have the SPECIAL attribute, or the profile must be within the scope of a group in which you have the group-SPECIAL attribute.
- If the profile is in the FILE or DIRECTORY class, the second qualifier of the profile name is your user ID.
- For a discrete profile, you are on the standard access list for the resource and you have ALTER authority.
- For a discrete profile, your current connect group (or, if list-of-groups checking is active, any group to which you are connected) is on the standard access list and has ALTER authority.
- For a discrete profile, the universal access authority is ALTER.

To specify the AT keyword, you must have READ authority to the DIRECT.node resource in the RRSFDATA class and a user ID association must be established between the specified node.userid pair(s).

To specify the ONLYAT keyword you must have the SPECIAL attribute, the userid specified on the ONLYAT keyword must have the SPECIAL attribute, and a user ID association must be established between the specified node.userid pair(s) if the user IDs are not identical.

When you are copying a list of authorized users from one resource profile to another, you must have sufficient authority, as described in the preceding list, to both of the resources.

Syntax

For the key to the symbols used in the command syntax diagrams, see "Syntax of RACF commands and operands" on page 9. The complete syntax of the PERMIT command is:

```
[subsystem-prefix][PERMIT | PE]

profile-name-1
   [ ACCESS(access-authority) | DELETE ]
   [ AT([node].userid ...) | ONLYAT([node].userid ...) ]
```
**PERMIT**

```plaintext
[ CLASS(profile-name-1-class) ]
[ FCLASS(profile-name-2-class) ]
[ FGENERIC ]
[ FROM(profile-name-2) ]
[ FVOLUME(volume-serial) ]
[ GENERIC ]
[ ID( {name ... | * } ) ]
[ RESET [ (ALL | STANDARD | WHEN) ]
[ VOLUME(volume-serial) ]
[ WHEN(
   [ APPCPORT( {partner-luname ... | * } ) ]
   [ CONSOLE( {console-id ... | * } ) ]
   [ CRITERIA( criteria-name ( {criteria-value | * } ) )]
   [ JESINPUT( {JES-input-device-name ... | * } ) ]
   [ PROGRAM( {program-name ... | * } ) ]
   [ SERVAUTH( {SERVAUTH-profile-name ... | * } ) ]
   [ SYSID( {system-identifier ... | * } ) ]
   [ TERMINAL( {terminal-id ... | * } ) ]
)]
```

For information on issuing this command as a RACF TSO command, refer to Chapter 3, “RACF TSO commands,” on page 15.

For information on issuing this command as a RACF operator command, refer to Chapter 4, “RACF operator commands,” on page 21.

**Parameters**

*subsystem-prefix*

Specifies that the RACF subsystem is the processing environment of the command. The subsystem prefix can be either the installation-defined prefix for RACF (1–8 characters) or, if no prefix has been defined, the RACF subsystem name followed by a blank. If the command prefix was registered with CPF, you can use the MVS command D OPDATA to display it or you can contact your RACF security administrator.

Only specify the subsystem prefix when issuing this command as a RACF operator command. The subsystem prefix is required when issuing RACF operator commands.

*profile-name-1*

Specifies the name of an existing discrete or generic profile whose access list you want to modify. You can specify only one profile.

This operand is required and must be the first operand following PERMIT.

If the name specified is a tape volume serial number that is a member of a tape volume set, the authorization assigned by this command applies to all the volumes in the volume set.

If the profile does not belong to the DATASET class, you must also specify CLASS.

Mixed-case profile names are accepted and preserved when CLASS refers to a class defined in the static class descriptor table with CASE=ASIS or in the dynamic class descriptor table with CASE(ASIS).
The AT and ONLYAT keywords are only valid when the command is issued as a RACF TSO command.

**AT**(*node*.userid ...)

Specifies that the command is to be directed to the node specified by *node*, where it runs under the authority of the user specified by *userid* in the RACF subsystem address space.

If *node* is not specified, the command is directed to the local node.

**ONLYAT**(*node*.userid ...)

Specifies that the command is to be directed only to the node specified by *node* where it runs under the authority of the user specified by *userid* in the RACF subsystem address space.

If *node* is not specified, the command is directed only to the local node.

**ACCESS**

**ACCESS**(*access-authority*)

Specifies the access authority you want to associate with the names that you identify on the ID operand. RACF sets the access authority in the standard access list.

If you specify WHEN, RACF sets the access authority in the conditional access list.

The valid access authorities are NONE, EXECUTE (for DATASET or PROGRAM class only), READ, UPDATE, CONTROL, and ALTER. If you need more information, see [z/OS Security Server RACF Security Administrator’s Guide](https://publib.boulder.ibm.com/infocenter/racfe/v2r1m0/index.jsp).

If you specify ACCESS and omit *access-authority*, the default value is ACCESS(READ).

If you specify the ID operand and omit both ACCESS and DELETE, the default value is ACCESS(READ).

If you specify both ACCESS and DELETE, RACF uses the last operand you specify.

**DELETE**

Specifies that you are removing the names you identify on the ID operand from an access list for the resource. RACF deletes the names from the standard access list.

If you specify WHEN, RACF deletes the names from the conditional access list.

If you specify the ID operand and omit both ACCESS and DELETE, the default value is ACCESS(READ).

If you specify both ACCESS and DELETE, RACF uses the last operand you specify.

**CLASS**(*profile-name-1-class*)

Specifies the name of the class to which *profile-name-1* belongs. The valid class names are DATASET and those classes defined in the class descriptor table. For a list of general resource classes defined in the class descriptor table supplied by IBM, see Appendix B, “Supplied RACF resource classes,” on page 659.

If you omit CLASS, the default is DATASET.
PERMIT

FCLASS(profile-name-2-class)
Specifies the name of the class to which profile-name-2 belongs. The valid class names are DATASET and those classes defined in the class descriptor table. For a list of general resource classes defined in the class descriptor table supplied by IBM, see Appendix B, “Supplied RACF resource classes,” on page 659.

If you specify FROM and omit FCLASS, RACF assumes that the class for profile-name-2 is same as the class for profile-name-1. This operand is valid only when you also specify the FROM operand; otherwise, RACF ignores it.

FGENERIC
Specifies that RACF is to treat profile-name-2 as a generic name, even if it is fully qualified (meaning that it does not contain any generic characters). This operand is only needed if profile-name-2 is a DATASET profile.

FROM(profile-name-2)
Specifies the name of the existing discrete or generic profile that contains the access lists RACF is to copy as the access lists for profile-name-1. If you specify FROM and omit FCLASS, RACF assumes that profile-name-2 is the name of a profile in the same class as profile-name-1.

Mixed-case profile names are accepted and preserved when FCLASS refers to a class defined in the static class descriptor table with CASE=ASIS or in the dynamic class descriptor table with CASE(ASIS).

If profile-name-2 contains a standard access list, RACF copies it to the profile you are changing. If profile-name-2 contains a conditional access list, RACF copies it to the profile you are changing.

RACF modifies the access list for profile-name-1 as follows:
• Authorizations for profile-name-2 are added to the access list for profile-name-1.

Note: The following conditional access list conditions are valid only for specific classes. Entries in the conditional access list of profile-name-2 for these conditions are copied to the conditional access list of profile-name-1 only if the condition is valid for the class of profile-name-1.
  – WHEN(SYSID) is valid only for the PROGRAM class. SYSID entries are copied only when the class of profile-name-1 is PROGRAM.
  – WHEN(PROGRAM) is valid only for data sets and the SERVAUTH class. PROGRAM entries are copied only when profile-name-1 is a data set profile or a SERVAUTH class profile.
  – WHEN(CRITERIA) is valid only for general resource classes (not data sets). CRITERIA entries are not copied when profile-name-1 is a data set profile.

• If a group or user appears in both lists, RACF uses the authorization granted in profile-name-1.
• If you specify a group or user on the ID operand and that group or user also appears in the profile-name-2 access list, RACF uses the authorization granted on the ID operand.

To specify FROM, you must have sufficient authority to both profile-name-1 and profile-name-2, as described under "Authorization required" on page 265.
FVOLUME(volume-serial)
Specifies the volume RACF is to use to locate profile-name-2. This is the volume on which the non-VSAM DASD data set, the tape data set, or the catalog for the VSAM data set resides.

If you specify FVOLUME and RACF does not find profile-name-2 on that volume, the command fails. If you omit this operand and profile-name-2 appears more than once in the RACF data set, the command fails.

FVOLUME is valid only when FCLASS either specifies or defaults to DATASET and when profile-name-2 specifies a discrete profile. Otherwise, RACF ignores FVOLUME.

GENERIC
Specifies that RACF is to treat profile-name-1 as a generic name, even if it does not contain any generic characters. This operand is only needed if profile-name-1 is a DATASET profile.

ID(name ... | *)
Specifies the user IDs and group names of RACF-defined users or groups whose authority to access the resource you are giving, removing, or changing.
If you omit this operand, RACF ignores the ACCESS and DELETE operands.

ID(*) can be used with standard or conditional access lists. You might specify ID(*) with a conditional access list, as follows:
PERMIT 'resource' ID(*) WHEN(PROGRAM(XYZ)) ACCESS(READ)

This command, depending on other environmental factors, may allow all RACF-defined users and groups READ access to the specified data set when executing program XYZ. RACF grants access to the data set, using the conditional access list, with the authority you specify on the ACCESS operand.
The value specified with ACCESS is used only if no more specific values are found. If you do not specify the ACCESS operand, or if you specify ACCESS without an access authority, RACF uses a default value of ACCESS(READ).
See z/OS Security Server RACF Security Administrator’s Guide for more information on program access to data sets.

For profiles in the FIELD class, you may also specify the value &RACUID for the name variable with the ID operand on the PERMIT command. When you enter this value on the PERMIT command, you allow all users access to the specified field or segment of their own user profiles.

RESET
RESET | RESET(ALL)
Specifies that RACF is to delete from the profile both the entire current standard access list and the entire current conditional access list.

RACF deletes both access lists before it processes any operands (ID and ACCESS or FROM) that create new entries in an access list. If you delete both access lists and specify FROM when profile-name-2 contains two access lists, the PERMIT command copies both access lists to profile-name-1. In any other situation, you cannot, on one PERMIT command, add entries to both access lists.

If you specify RESET and do not specify ALL, STANDARD, or WHEN, the default value is RESET(ALL).

If you specify RESET or RESET(ALL), add entries, and omit WHEN, RACF deletes both access lists, then adds entries to the standard access list.
If you specify RESET or RESET(ALL), add entries, and specify WHEN, RACF deletes both access lists, then adds entries to the conditional access list.

For profiles that include two access lists, use RESET and RESET(ALL) carefully. Unless you are copying both lists from another profile, it is a good practice to use RESET(STANDARD) to maintain the standard access list and RESET(WHEN) to maintain the conditional access list.

**RESET(STANDARD)**

Specifies that RACF is to delete the entire current standard access list from the profile.

If you specify RESET(STANDARD) with ID and ACCESS or with FROM, RACF deletes the current standard access list from the profile before it adds the new names.

If you specify RESET(STANDARD) with ID and DELETE, RACF ignores RESET(STANDARD) and deletes only the names that you specify.

If you specify RESET(STANDARD) without ID and ACCESS, or without FROM, the resulting standard access list is empty. An empty standard access list means that, for a general resource or a group data set profile, you must be the owner or have the SPECIAL attribute, or the profile must be within the scope of a group in which you have the group-SPECIAL attribute, in order to update the access list again.

For a DATASET profile, an empty conditional access list means that no users or groups can access the data set by executing a program.

**RESET(WHEN)**

Specifies that RACF is to delete the entire current conditional access list from the profile.

If you specify RESET(WHEN) with ID and ACCESS or with FROM, RACF deletes the current conditional access list from the profile before it adds the new names.

If you specify RESET(WHEN) with ID, DELETE, and WHEN, RACF ignores RESET(WHEN) and deletes only the names that you specify.

If you specify RESET(WHEN) without ID and ACCESS, or without FROM, the resulting conditional access list is empty.

**VOLUME(volume-serial)**

Specifies the volume on which the tape data set, the non-VSAM DASD data set, or the catalog for the VSAM data set resides.

If you specify VOLUME and volume-serial does not appear in the profile for the data set, the command fails.

If you omit VOLUME and the data set name appears more than once in the RACF data set, the command fails.

This operand is valid only for CLASS(DATASET). RACF ignores it for all other CLASS values.

If profile-name-1 is a generic profile, RACF ignores this operand.

**WHEN(APPCPORT(partner-luname ... | *))**

Specifies that the indicated users or groups have the specified access authority when executing commands and jobs originating from the specified partner LU.
PERMIT

Specify one or more LU names. No generic names or profile names are supported.

WHEN(APPCPORT(*)) deletes all APPCPORT entries for the specified users or groups. It is valid only with the DELETE operand.

**Note:** The LU name must be qualified with the network name if the installation is using the network qualified names feature on the APPC connection. For more information, refer to [z/OS Security Server RACF Security Administrator's Guide](https://www.ibm.com).  

WHEN(CONSOLE(console-id ... | *))  
Specifies that the indicated users or groups have the specified access authority when executing commands and jobs originating from the specified system console.

Specify one or more console identifiers. No generic names or profile names are supported.

WHEN(CONSOLE(*)) deletes all CONSOLE entries for the specified users or groups. It is valid only with the DELETE operand.

WHEN(CRITERIA(criteria-name (criteria-value | *)))  
Specifies that the indicated users or groups have the specified access authority when they are defined in an application that uses the specified criteria. Applications, such as DB2®, can execute the RACROUTE REQUEST=FASTAUTH request to check user and group authority to access a resource associated with a particular criteria, such as a DB2 role.

**Important:** Specify the same criteria name and value that the application specifies on the RACROUTE REQUEST=FASTAUTH request. For details about valid criteria names and values, see your application documentation. For information about RACROUTE, see [z/OS Security Server RACROUTE Macro Reference](https://www.ibm.com).

The criteria-name is a string of 1–8 characters. The string can contain any combination of A–Z, 0–9, # (X'7B'), $ (X'5B'), or @ (X'7C'). It must not contain blanks. Lowercase alphabetic characters in the criteria-name are translated to upper case.

The criteria-value is a string of 1–235 characters of any combination. If the criteria-value consists of a single asterisk (*), you can optionally enclose it in single quotation marks. If the criteria-value contains blanks or other special characters, you must enclose the entire string in single quotation marks.

When the criteria-value is enclosed in single quotation marks, the following rules apply:
- The string must contain at least one non-blank character.
- The string must not contain blanks between the last character and the ending quote.
- If a single quotation mark is intended to be part of the criteria-value, use two single quotation marks together for each single quotation mark within the string, and enclose the entire string within single quotation marks.

The criteria-value is stored in the RACF database exactly as you specify it:
- Both uppercase and lowercase characters are preserved in the case in which they are specified.
- Leading blanks are preserved when the string is quoted.

WHEN(CRITERIA(SQLROLE(DB2-role-name)))  
Beginning with DB2 Version 9, you can authorize conditional access to DB2...
resources for users and groups associated (in DB2) with a DB2 role by specifying SQLROLE as the criteria-name and a DB2 role name as the criteria-value. Specify DB2-role-name to match a DB2-defined role name. (For more information about using DB2 roles, see the DB2 Version 9 publication library.)

**Example:** WHEN(CRITERIA(SQLROLE(TELLER)))

WHEN(CRITERIA(SQLROLE('))) and WHEN(CRITERIA(SQLROLE(‘’))) delete all SQLROLE CRITERIA entries for the specified users or groups.

WHEN(JESINPUT(JES-input-device-name ... I *))

Specifies that the indicated users or groups have the specified access authority when entering the system through the specific JES input device.

Specify one or more device names. No generic names or profile names are supported.

WHEN(JESINPUT(‘’)) deletes all JESINPUT entries for the specified users or groups. It is valid only with the DELETE operand.

WHEN(PROGRAM(program-name ... I *))

Specifies that you want to create or delete entries in the conditional access list of the specified data set or SERVAUTH profile. This operand applies only to resources in the data set and SERVAUTH classes.

Specify one or more program names. No generic names or profile names are supported.

For example, if you enter the following command:

```PERMIT 'XXX.YYY' ID(SMITH) ACCESS(READ) WHEN(PROGRAM(ABC))```

RACF allows user SMITH READ access to the data set protected by profile XXX.YYY when executing program ABC. RACF grants access, through the conditional access list, with the authority you specify on the ACCESS operand. If you do not specify the ACCESS operand, or if you specify ACCESS without an access authority, RACF uses a default value of ACCESS(READ).

See [z/OS Security Server RACF Security Administrator's Guide](http://pic.dhe.com/boe/bill/97-003.pdf) for more information on data set access and program access to SERVAUTH resources when program control is active.

WHEN(PROGRAM) affects only users and groups specified on the ID operand; it has no effect on names copied from a standard access list in another profile (using the FROM operand). Thus, you can copy a standard access list from another profile that contains only a standard access list and add or delete names in the conditional access list on a single PERMIT command.

To delete an entry from the conditional access list of a data set profile, issue the PERMIT command as follows:

```PERMIT 'XXX.YYY' ID(JONES) DELETE WHEN(PROGRAM(ABC))```

When you issue this command, RACF no longer allows user JONES access to the data set protected by profile XXX.YYY when executing program ABC. If you specify WHEN(PROGRAM(‘’)) with DELETE, RACF deletes all program names for each user or group specified on the ID operand.

See also the description of the ID operand.

WHEN(PROGRAM(‘’)) deletes all PROGRAM entries for the specified users or groups. It is valid only with the DELETE operand.

WHEN(SERVAUTH(SERVAUTH-profile-name ... I *))
WHEN(SERVAUTH(SERVAUTH-profile-name ...))
Specifies that the indicated users or groups have the specified access authority when using an IP address protected by the named SERVAUTH profile. The profile name may be generic; however, it must match exactly the name of a profile to allow access.

Guideline: Use careful consideration before specifying the SERVAUTH profile name * on the RDEFINE and PERMIT WHEN(SERVAUTH(...)) commands. The SERVAUTH profile name * cannot be removed from the conditional access list without deleting all SERVAUTH entries for the specified users or groups. Instead, we recommend that you create the profile ** in the SERVAUTH class. Then use the ** profile name for the conditional access list.

WHEN(SERVAUTH(+) )
Deletes all SERVAUTH entries for the specified users or groups when specified with the DELETE operand.

WHEN(SYSID(system-identifier ... | * ))
Specifies that the indicated users or groups have the specified access authority when loading this controlled program on the specified system.

Specify one or more system identifiers. No generic names or profile names are supported.

This operand applies only to resources in the PROGRAM class. The system-identifier is the 4-character value specified for the SID parameter of the SMFPRMxx member of SYS1.PARMLIB. See z/OS MVS Initialization and Tuning Reference for additional information on SMFPRMxx.

WHEN(SYSID( + )) deletes all SYSID entries for the specified users or groups. It is valid only with the DELETE operand.

WHEN(TERMINAL(terminal-id ... | * ))
Specifies that the indicated users or groups have the specific access authority when logged on to the named terminal.

Specify one or more terminal identifiers. No generic names or profile names are supported.

WHEN(TERMINAL( + )) deletes all TERMINAL entries for the specified users or groups. It is valid only with the DELETE operand.

Examples
Table 19. PERMIT examples

Example 1

Operation  User WJE10 wants to give UPDATE access authority to data set WJE10.DEPT2.DATA to all the users in the group RESEARCH. Data set WJE10.DEPT2.DATA is protected by a discrete profile.

Known  User WJE10 and group RESEARCH are RACF-defined.

Data set WJE10.DEPT2.DATA is RACF-defined.

User WJE10 wants to issue the command as a RACF TSO command.

Command  PERMIT 'WJE10.DEPT2.DATA' ID(RESEARCH) ACCESS(UPDATE)

Defaults  CLASS(DATASET)
Table 19. PERMIT examples (continued)

Example 2

**Operation** User WRH0 wants to give all users authorized to access the data set RESEARCH.PROJ01.DATA on volume DASD22 the authority to access RESEARCH.PROJ01.DATA on volume DASD11. User WRH0 also wants to give user AEH10 READ authority to RESEARCH.PROJ01.DATA.

**Known** User WRH0 has ALTER access to both RESEARCH.PROJ01.DATA data sets. Both data sets are protected by discrete profiles. User WRH0 wants to issue the command as a RACF TSO command.

**Command** PERMIT 'RESEARCH.PROJ01.DATA' ID(AEH10) FROM('RESEARCH.PROJ01.DATA') VOLUME(DASD11) FVOLUME(DASD22)

**Defaults** ACCESS(READ) CLASS(DATASET) FCLASS(DATASET)

Example 3

**Operation** User LAB2 wants to delete user MMC02’s access to tape volume TAP8X.

**Known** User LAB2 is the owner of the profile for tape volume TAP8X. User LAB2 wants to issue the command as a RACF operator command, and the RACF subsystem prefix is @.

**Command** @PERMIT TAP8X CLASS(TAPEVOL) ID(MMC02) DELETE

**Defaults** None.

Example 4

**Operation** User ADM1 wants to delete the existing standard access list from the discrete profile protecting the data set SALES.EUROPE.ABC, then copy the standard access list from the generic profile SALES.*.ABC to the discrete profile for SALES.EUROPE.ABC User ADM1 wants to direct the command to run under the authority of user THB11.

**Known** User THB11 has the SPECIAL attribute. SALES.EUROPE.ABC is in the DATASET class. User ADM1 wants to issue the command as a RACF TSO command.

ADM1 and THB111 have an already established user ID association.

**Command** PERMIT 'SALES.EUROPE.ABC' FROM('SALES.*.ABC') RESET(STANDARD) AT(.THB11)

**Defaults** CLASS (DATASET) FCLASS(DATASET)

Command direction defaults to the local node.

Example 5

**Operation** User ADM1 wants to replace the conditional access list in the discrete profile that protects the data set SALES.EUROPE.ABC. Two users, TH01 and TH03, are to be allowed to update the data set when executing the program named FUTURE.

**Known** User ADM1 has the SPECIAL attribute. Users TH01 and TH03 are defined to RACF. The program FUTURES has been defined to RACF as a controlled program. User ADM1 wants to issue the command as a RACF TSO command.

**Command** PERMIT 'SALES.EUROPE.ABC' RESET(WHEN) ID(TH01 TH03) ACCESS(UPDATE) WHEN(PROGRAM(FUTURES))

**Defaults** CLASS(DATASET)

Example 6

**Operation** User ADM1 wants to control the access of shared user IDs PUBLIC and RESELL to data sets containing sales data. All users working within the company need access to sales data along with RESELL, but PUBLIC cannot have access.

**Known** User ADM1 has the SPECIAL attribute. User IDs PUBLIC and RESELL have the RESTRICTED attribute. SALES RESELL.* is a generic data set with a UACC(READ).

**Command** PERMIT 'SALES.RESELL.*' ID(RESELL) ACCESS(READ)

**Defaults** None.
Example 7

**Operation**  Rui wants to authorize user JEAN to alter a DB2 table owned by ZHAOHUI only when JEAN is assigned in DB2 to the role called TELLER.

**Known**  Rui has the SPECIAL attribute. A general resource called DSN.ZHAOHUI.TABLE.ALTER is defined in the MDSNTB class with UACC(NONE). The user JEAN is assigned in DB2 to the role called TELLER. The installation uses the RACF access control module (ACM) with DB2. The ACM is configured for multiple-subsystem scope and the DB2 subsystem is operational.

**Command**  PERMIT DSN.ZHAOHUI.TABLE.ALTER CLASS(MDSNTB) ID(JEAN) ACCESS(READ) WHEN(CRITERIA(SQLROLE(TELLER)))

**Defaults**  None.
The RACDCERT command has numerous functions. Each function of the RACDCERT command is documented in a separate topic that contains its purpose, authorization required, syntax, and other specific information.

For details about each RACDCERT function, see the following topics:

- RACDCERT ADD (Add certificate) on page 283
- RACDCERT ADDRING (Add key ring) on page 293
- RACDCERT ADTOKEN (Add token) on page 295
- RACDCERT ALTER (Alter certificate) on page 297
- RACDCERT ALTMAP (Alter mapping) on page 300
- RACDCERT BIND (Bind certificate to token) on page 303
- RACDCERT CHECKCERT (Check certificate) on page 307
- RACDCERT CONNECT (Connect a certificate to key ring) on page 310
- RACDCERT DELETE (Delete certificate) on page 314
- RACDCERT DELMAP (Delete mapping) on page 317
- RACDCERT DELRING (Delete key ring) on page 320
- RACDCERT DETOKEN (Delete token) on page 322
- RACDCERT EXPORT (Export certificate package) on page 324
- RACDCERT GENCERT (Generate certificate) on page 328
- RACDCERT GENREQ (Generate request) on page 340
- RACDCERT IMPORT (Import certificate) on page 343
- RACDCERT LIST (List certificate) on page 348
- RACDCERT LISTMAP (List mapping) on page 355
- RACDCERT LISTRING (List key ring) on page 358
- RACDCERT LISTTOKEN (List token) on page 361
- RACDCERT MAP (Create mapping) on page 364
- RACDCERT REKEY (Rekey certificate) on page 371
- RACDCERT REMOVE (Remove certificate from key ring) on page 377
- RACDCERT ROLLOVER (Rollover certificate) on page 379
- RACDCERT UNBIND (Unbind certificate from token) on page 382

Purpose

Use the RACDCERT command to install and maintain digital certificates, key rings, and digital certificate mappings in RACF. RACDCERT should be used for all maintenance of profiles in the DIGTCERT, DIGTRING, and DIGTNMAP classes.

The RACDCERT command is a RACF TSO command used to:

- List information about the certificates for a specified RACF-defined user ID, or your own user ID.
- Add a certificate and associate it with a specified RACF-defined user ID, or your own user ID, and set the TRUST status.
- Check to see if a certificate has been defined to RACF.
- Alter the TRUST status or label for a certificate.
- Delete a certificate.
- List a certificate contained in a data set and determine if it is associated with a RACF-defined user ID.
- Add or remove a certificate from a key ring.
- Create, delete, or list a key ring.
- Generate a public/private key pair and certificate, replicate a digital certificate with a new public/private key pair, or retire the use of an existing private key.
- Write (export) a certificate or certificate package to a data set.
RACDCERT (common)

- Create a certificate request.
- Create, alter, delete, or list a certificate name filter (user ID mapping).
- Add, delete, or list a z/OS PKCS #11 token.
- Bind a certificate to a z/OS PKCS #11 token.
- Remove (unbind) a certificate from a z/OS PKCS #11 token.
- Import a certificate (with its private key, if present) from a z/OS PKCS #11 token and add it to RACF.

RACF signs its certificates using a signature algorithm based on the SHA-1 standard.

For increased security and performance for signature verifications, RACF uses an exponent value of 65537 for each key it generates with the RSA algorithm.

Authorization required

To issue the RACDCERT command, you must have sufficient authority for the specific RACDCERT function. This authority may include one of the following authorities, or both, depending on the command function:

- SPECIAL, or sufficient authority to the appropriate resource in the FACILITY class.
- Sufficient ICSF authority to the appropriate resource in the CRYPTOZ class.

For details about CRYPTOZ authorities, see z/OS Cryptographic Services ICSF Administrator’s Guide.

For authorization details about each RACDCERT function, see the “Authorization required” topic for the RACDCERT function.

Controlling the use of RACDCERT: Effective use of RACDCERT requires that its privileges be carefully controlled. However, end users and application administrators should be allowed some flexibility in defining their security characteristics.

Guidelines:

- Give the authority to add certificate authorities to only a small set of trusted people.
- End users need to add, delete, and modify the contents of their own key rings and to add, delete, and alter their own certificates.
- Help desk personnel need to list certificates and key rings.

Example: Figure 36 on page 281 lists sample commands to implement one method of controlling RACDCERT access according to these guidelines. In the example shown, the system administrators (who are the only ones to add, alter, or delete certificate-authority certificates or site certificates) are in the WEBADMIN group and the help desk personnel are in the HELPDESK group.

Syntax

For details about syntax and parameters for each RACDCERT function, see the “Syntax” and “Parameters” subtopics in the topic for each RACDCERT function.

If you specify more than one RACDCERT function, only the last specified function is processed. Extraneous keywords that are not related to the function being performed are ignored.

If you do not specify a RACDCERT function, LIST is the default function.
UTF-8 and BMP character restrictions

You can include UTF-8 and BMP characters in certificate names with the following restrictions:

- You can specify certificate names that include UTF-8 and BMP characters only when they are part of an encoded certificate or certificate request that is stored in an MVS data set, and you specify the data set name with your RACDCERT command.

- Do not use keyboard entries (including cut-and-paste methods) to specify UTF-8 and BMP characters as command-line input. UTF-8 or BMP characters specified at the command line might be incorrectly processed, although you might receive no input error.

- Any UTF-8 or BMP character that does not map to the IBM-1047 code page is represented by six characters in the \texttt{U+n\texttt{n}\texttt{nnn}} format, where \texttt{n\texttt{nnn}} is the hexadecimal form of the Unicode code point for the UTF-8 or BMP character. For example, the Euro symbol (€) is represented as \texttt{U+20AC}.

For a sample listing of a certificate that contains information that includes an unmapped character, see \texttt{Figure 43 on page 354}.

When one unmapped UTF-8 or BMP character is represented by six characters, the additional five characters of length might affect the processing of certain certificates, such as in the following cases:

- When the issuer’s distinguished name is lengthy and contains one or more unmapped UTF-8 or BMP characters, the resulting profile name for the certificate might exceed the allowable length for a profile name. If this occurs, the RACDCERT ADD or GENCERT command fails and the certificate is not added.

- When RACF generates a default label for a certificate extracted from a PKCS #12 package during RACDCERT ADD processing and the certificate’s friendly name contains one or more unmapped UTF-8 or BMP characters, the resulting label might exceed 32 characters. If this occurs, RACF truncates the label.

DEBUG keyword

Add the DEBUG keyword when you issue the RACDCERT command to obtain additional diagnostic messages for failures related to encryption calls, and RACF-invoked ICHEINTY ALTER, RACROUTE REQUEST=EXTRACT, and RACROUTE REQUEST=DEFINE calls.

The content of these additional diagnostic messages are not documented in the RACF publication library.

If you report a problem to the IBM Support Center, use the DEBUG keyword to gather diagnostic information.

ICSF considerations

RACDCERT processing makes use of ICSF services. If your installation has established access control over ICSF services, then the issuers of RACDCERT will need to be granted READ authority to ICSF services according to \texttt{Table 20 on page 279}. 
Table 20. ICSF services (resources in the CSFSERV class) used during RACDCERT command processing

<table>
<thead>
<tr>
<th>RACDCERT command</th>
<th>Keywords</th>
<th>ICSF service (resource in the CSFSERV class)</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADDTOKEN</td>
<td>—</td>
<td>CSF1TRC</td>
<td>—</td>
</tr>
<tr>
<td>BIND</td>
<td>—</td>
<td>CSF1GAV, CSF1SAV, CSF1TRC, and CSF1TRL</td>
<td>—</td>
</tr>
<tr>
<td>DELETE</td>
<td>—</td>
<td>CSFPKRD</td>
<td>Only if the certificate being deleted has an ICSF or PCICC key</td>
</tr>
<tr>
<td>DELTOKEN</td>
<td>—</td>
<td>CSF1GAV, CSF1TRD, and CSF1TRL</td>
<td>—</td>
</tr>
<tr>
<td>GENCERT or REKEY</td>
<td>—</td>
<td>CSFRNG</td>
<td>Required only when ICSF is active</td>
</tr>
<tr>
<td>GENCERT or REKEY</td>
<td>PCICC</td>
<td>CSFDSDG and CSFPKG</td>
<td>—</td>
</tr>
<tr>
<td>GENCERT, REKEY, or ADD</td>
<td>ICSF or PCICC</td>
<td>CSFPKI, CSFPKRC, CSFPKRR, CSFPKRW, and CSFPKX.</td>
<td>—</td>
</tr>
<tr>
<td>GENCERT FROMICSF</td>
<td>—</td>
<td>CSFPKX</td>
<td>—</td>
</tr>
<tr>
<td>GENCERT SIGNWITH</td>
<td>—</td>
<td>CSFDSDG, CSFPKRR, and CSFPKX</td>
<td>Only if the signing certificate has an ICSF or PCICC key.</td>
</tr>
<tr>
<td>IMPORT</td>
<td>—</td>
<td>CSF1GAV and CSF1TRL</td>
<td>—</td>
</tr>
<tr>
<td>LISTTOKEN</td>
<td>—</td>
<td>CSF1GAV and CSF1TRL</td>
<td>—</td>
</tr>
<tr>
<td>UNBIND</td>
<td>—</td>
<td>CSF1GAV, CSF1TRD, and CSF1TRL</td>
<td>—</td>
</tr>
</tbody>
</table>

When the ICSF or PCICC keyword is specified for ADD, GENCERT, IMPORT, or REKEY, RACF stores the key in the ICSF PKA key data set (PKDS). If no PKDS label is specified for the ICSF key, RACF generates a default label in the format IRR.DIGTCERT.certificate-owner.cvtname.ebcdic-stck-value, where certificate-owner is the owning user ID, cvtname is the system name as taken from the CVT, and ebcDIC-stck-value is an EBCDIC version of the current store-clock value. If the key is associated with a certificate-authority certificate, user ID is set to CERTIFAUTH. If the key is associated with a site certificate, then user ID is set to SITECERTIF.

If your installation has established access control over keys stored in ICSF, the issuers of the RACDCERT command must have READ access authority to ICSF keys by label. Because the specific label values might be hard to determine, generic profiles are suggested according to Table 21 on page 280, based on the issued RACDCERT function and the attributes of the ICSF key.
### Table 21. Suggested generic profiles (resources in the CSFKEYS class) that authorize access to ICSF keys based on RACDCERT function and key attributes

<table>
<thead>
<tr>
<th>RACDCERT command</th>
<th>Keywords</th>
<th>Suggested generic profile (CSFKEYS class resource)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADD, GENCERT or ADD, GENCERT or REKEY for ID(cert-owner)</td>
<td>ICSF or PCICC</td>
<td>IRR.DIGTCERT.cert-owner.*</td>
</tr>
<tr>
<td>ADD, GENCERT or ADD, GENCERT or REKEY for CERTAUTH</td>
<td>ICSF or PCICC</td>
<td>IRR.DIGTCERT.CERTIFAUTH.*</td>
</tr>
<tr>
<td>ADD, GENCERT or ADD, GENCERT or REKEY for SITE</td>
<td>ICSF or PCICC</td>
<td>IRR.DIGTCERT.SITECERTIF.*</td>
</tr>
<tr>
<td>GENCERT for GENCERT</td>
<td>SIGNWITH(LABEL('label'))</td>
<td>IRR.DIGTCERT.cert-owner.*</td>
</tr>
<tr>
<td>ID(cert-owner)</td>
<td>SIGNWITH(CERTAUTH LABEL('label'))</td>
<td>IRR.DIGTCERT.CERTIFAUTH.*</td>
</tr>
<tr>
<td>GENCERT</td>
<td>SIGNWITH(SITE LABEL('label'))</td>
<td>IRR.DIGTCERT.SITECERTIF.*</td>
</tr>
<tr>
<td>GENREQ or DELETE for GENREQ or DELETE for ID(cert-owner)</td>
<td>—</td>
<td>IRR.DIGTCERT.cert-owner.*</td>
</tr>
<tr>
<td>ID(cert-owner)</td>
<td>—</td>
<td>IRR.DIGTCERT.CERTIFAUTH.*</td>
</tr>
<tr>
<td>GENREQ or DELETE for SITE</td>
<td>—</td>
<td>IRR.DIGTCERT.SITECERTIF.*</td>
</tr>
</tbody>
</table>

Additionally, the user ID assigned to an application, such as System SSL, that uses certificates stored in RACF key rings also needs READ authority to ICSF keys by label. Because the specific label values might be hard to determine, generic profiles are suggested according to Table 22 based on the CONNECT attributes of the ICSF key.

### Table 22. Suggested generic profiles (resources in the CSFKEYS class) that authorize access to ICSF keys based on CONNECT attributes

<table>
<thead>
<tr>
<th>RACDCERT CONNECT keywords used to populate the key ring</th>
<th>Suggested generic profile (CSFKEYS class resource)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CONNECT(LABEL('label')) for ID(cert-owner)</td>
<td>IRR.DIGTCERT.cert-owner.*</td>
</tr>
<tr>
<td>CONNECT(CERTAUTH LABEL('label') USAGE(PERSONAL))</td>
<td>IRR.DIGTCERT.CERTIFAUTH.*</td>
</tr>
<tr>
<td>CONNECT(SITE LABEL('label-name') USAGE(PERSONAL))</td>
<td>IRR.DIGTCERT.SITECERTIF.*</td>
</tr>
</tbody>
</table>

Sufficient ICSF authority for the following command functions is controlled using resources in the CRYPTOZ class. If the CSFSERV class is active, sufficient ICSF authority for the following command functions might also be required. For authorization details, see [z/OS Cryptographic Services ICSF Administrator’s Guide](https://www.ibm.com/docs/en/zos-crypto-services).

- ADDTOKEN
- BIND
- DELTOKKEN
- IMPORT
- LISTTTOKEN
- UNBIND.

### Examples
Figure 36. Controlling access to RACDCERT functions
RACDCERT (common)
RACDCERT ADD (Add certificate)

Purpose

Use the RACDCERT ADD command to define a digital certificate using a certificate or certificate package contained in the specified data set.

Each user ID can be associated with more than one digital certificate but they must be added individually. The specified data set should contain only one digital certificate. The command reads the certificate from the data set, updates the user’s profile, and creates the DIGTCERT profile.

See “UTF-8 and BMP character restrictions” on page 278 for information about how UTF-8 and BMP characters in certificate names and labels are processed by RACDCERT functions.

Processing details

Re-adding certificates:
If the certificate being added already exists in the RACF database, RACF will accept the certificate and refresh its stored information if all of the following conditions are true:

- The certificate is being added to the same user ID, SITE, or CERTAUTH as before.
- The label specified for the certificate matches the old value or no label is specified.

Otherwise, an informational message is issued, and the certificate is not re-added.

Renewing certificates:
The ADD keyword also supports certificate replacement, for cases of certificate renewal and fulfillment by an external certificate authority. When a certificate is replaced, all existing information is updated to reflect the new certificate. This includes key ring connections.

The certificate in the RACF database is replaced if the following conditions are true.

- If the existing certificate has a private key associated with it:
  - The certificate is being added to the same user ID, SITE, or CERTAUTH as before.
  - The certificate being added is not a duplicate. (You are not re-adding the certificate.)
  - The certificate being added is not expired.
  - The certificate being added has the same public key as the existing certificate.

- If the existing certificate does not have a private key associated with it:
  - The certificate is being added to the same user ID, SITE, or CERTAUTH as before.
  - The certificate being added is not a duplicate. (You are not re-adding the certificate.)
  - The certificate being added is not expired.
  - The certificate being added has a later expiration date than that of the existing certificate.
RACDCERT ADD

- The certificate being added has the same subject's distinguished name, issuer's distinguished name, and public key as the existing certificate.

Note: When a certificate is being replaced, a new label can be specified.

Supported certificate package formats:
The certificate package must be in one of the following formats:
1. A single BER encoded X.509 certificate.
2. A single Base64 encoded X.509 certificate.
3. A Privacy Enhanced Mail (PEM) encoded X.509 certificate. If the input is in this format, only the Originator Certificate is used.
4. One or more X.509 certificates contained within a PKCS #7 DER encoding package.
5. One or more X.509 certificates and private keys contained within a PKCS #12 DER encoding package. If the input is in this format, all certificates are processed but only the first user private key is used. PKCS #12 is also known as Private Information Exchange (PFX). The obsolete PFX V0.02 standard is not supported.

Details about adding new certificates:
The following are additional details regarding RACDCERT's certificate processing:
1. All fields as defined for X.509 version 1 certificates must be present and must have a length greater than zero (non-null).
2. The combined length of the serial number and issuer's distinguished name cannot exceed 246 bytes, the maximum length for the name of a DGTTCERT profile.
3. X.509 certificates with version numbers greater than 3 are not supported.
4. Noncritical extensions are ignored. Critical extensions that are supported include:
   • keyUsage—{252 91 5}
   • basicConstraints—{252 91 9}
   • subjectAltname—{252 91 7}
   • issuerAltName—{252 91 8}
   • certificatePolicies—{252 93 2}
   • policyMappings—{252 93 3}
   • policyConstraints—{252 93 6}
   • nameConstraints—{252 93 0}
   • extKeyUsage—{252 93 7}
   • hostIdMapping—{131 8021 81}
   • subjectKeyIdentifier—{252 91 4}
   • authorityKeyIdentifier—{252 93 5}
5. Subject and issuer names can contain only the following string types:
   • UTF8—TAG 12
   • PRINTABLESTRING—TAG 19
   • T61STRING—TAG 20
   • IA5STRING—TAG 22
   • VISIBLESTRING—TAG 26
   • GENERALSTRING—TAG 27
   • BMPString—TAG 30
6. Because certificates can be encoded differently, be aware when transporting the different certificate encodings to and from z/OS and OS/390 systems. Both the BER encoded X.509 and PKCS #7 formats are binary formats and must be transported in their exact binary format. For example, binary formats, such as BER and X.509, cannot have any ASCII to EBCDIC translation performed on them, therefore, they must be transported in their exact binary format. In contrast, text formats, such as PEM and Base64, must be transported as text. When transporting for an ASCII system, the ASCII to EBCDIC translation must be performed for the PEM format and Base64 format certificate.

PKCS #12 format details:
PKCS #12 certificate packages can be processed. These certificates are encrypted with a password when received and must be decrypted with the same password before being added to the RACF database. For additional information, see the PASSWORD keyword on page 289.

When adding a certificate package that contains a private key, if ICSF is being used to store private keys and no label is specified for the ICSF key, ADD generates a default label in the format IRR.DIGTCERT.certificate-ownerv cvtsname.ebcdic-stck-value, where certificate-owner is the owning user ID, cvtsname is the system name, as taken from the CVT, and ebcdisch-stck-value is an EBCDIC version of the current store-clock value. See "PKCS #7 and PKCS #12 processing details" for information on how the multiple certificates are processed.

PKCS #7 and PKCS #12 processing details:
The ADD function of RACDCERT can accept PKCS #7 and PKCS #12 certificate packages. For PKCS #7, if there is more than one certificate in the package, the set consisting of the second through last certificate is the hierarchy of CAs. For PKCS #12, every certificate in the package other than the first one that has a 'local key ID' that matches the first private key in the package, is considered to be a CA certificate. If the command issuer is authorized to add CERTAUTH certificates, the CA certificates will be sorted to determine the hierarchy chain. The resulting set is then added to the CERTAUTH category in the hierarchy order (top CA down to lowest CA). Thus each certificate in the package may be verified using its previously added parent. If the command issuer is not authorized to add CERTAUTH certificates, an informational message will be issued. In either case, processing will then continue with the first certificate in the package (the end-entity certificate). For each CERTAUTH certificate in the PKCS #7 or PKCS #12 package, the following list of rules will apply to determine the trust status. The list is in priority order. For rules that conflict, the first matching rule wins:

Rules:
1. If the certificate is already defined to RACF with status HIGHTRUST, the certificate retains its HIGHTRUST status.
2. The trust status specified by the command issuer will apply to the top CA certificate. This primes the chain with a trust value which may be inherited down. (See the next rule.) The HIGHTRUST keyword is ignored if the target user ID on the ADD is not CERTAUTH (irrcerta).
3. For all lower CA certificates in the package and for the top CA certificate when no trust status is specified, the trust status is determined dynamically as follows:
   a. If NOTRUST is specified by the command issuer, the certificate is added with status NOTRUST.
b. If the certificate has one or more of the following inconsistencies, the certificate is added with NOTRUST status:

1) The certificate is expired.

2) The certificate has an incorrect date range relative to the issuing CA certificate. (The validity period is not completely contained within the validity period of the issuing CA certificate).

3) The issuer of the certificate is missing from the package and is not already installed under CERTAUTH.

4) The certificate has an unknown signature algorithm.

c. If no inconsistencies are detected, the certificate is added and inherits the trust status of its parent. If the certificate’s parent has not previously been added (either as a part of this package or otherwise), the certificate is added with TRUST status if it is self-signed, NOTRUST status if it is not self-signed. If the self-signed certificate has already been added, its trust status is not changed.

4. HIghTRUST will be inherited from the parent as per the previous rule only if the target user ID on the ADD is CERTAUTH (irrcerta) and HIGHERTRUST was specified on the command. In all other cases, HIghTRUST reverts to TRUST when inheriting from the parent.

5. The LABEL value will not be used. The label will be generated.

The authority required to add the CERTAUTH certificates from a PKCS #7 or PKCS #12 package is the same authority required to add CERTAUTH certificates directly, either CONTROL authority to IRR.DIGTCERT.ADD in the FACILITY class or RACF SPECIAL.

Note: PKCS #7 and PKCS #12 add error processing that has no backout support. If a terminating error is encountered during processing, any CERTAUTH certificates previously added are not removed. Unless otherwise stated in the error message description, any error messages issued are relative to the certificate where the error occurred. This may be the end-entity certificate or one of the CERTAUTH certificates.

Issuing options

The following table identifies the eligible options for issuing the RACDCERT ADD command:

<table>
<thead>
<tr>
<th>As a RACF TSO command?</th>
<th>As a RACF operator command?</th>
<th>With command direction?</th>
<th>With automatic command direction?</th>
<th>From the RACF parameter library?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>No</td>
<td>No. (See rules.)</td>
<td>No. (See rules.)</td>
<td>No</td>
</tr>
</tbody>
</table>

Rules:
- The RACDCERT command cannot be directed to a remote system using the AT or ONLYAT keyword.
- The updates made to the RACF database by RACDCERT are eligible for propagation with automatic direction of application updates based on the RRSFDATA profiles AUTODIRECT.target-node.DIGTCERT.APPL and AUTODIRECT.target-node.DIGTRING.APPL, where target-node is the remote node to which the update is to be propagated.

Authorization required

To issue the RACDCERT ADD command, you must have the following authorizations:
1. The SPECIAL attribute, or sufficient authority to the IRR.DIGTCERT.ADD resource in the FACILITY class for your intended purpose, as shown in Table 23.

2. READ access to the data set that contains the certificate you are adding.

### Table 23. Authority required for the RACDCERT ADD function

<table>
<thead>
<tr>
<th>IRR.DIGTCERT.ADD</th>
<th>Access level</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>READ</td>
<td>Add a certificate to your own user ID.</td>
<td></td>
</tr>
<tr>
<td>UPDATE</td>
<td>Add a certificate for another user ID.</td>
<td></td>
</tr>
<tr>
<td>CONTROL</td>
<td>Add a SITE or CERTAUTH certificate.</td>
<td></td>
</tr>
</tbody>
</table>

### Activating your changes

If the DIGTCERT or DIGTRING class is RACLISTed, refresh the classes to activate your changes.

**Example:**

```
SETROPTS RACLIST(DIGTCERT, DIGTRING) REFRESH
```

### Related commands

- To alter a certificate, see RACDCERT ALTER.
- To delete a certificate, see RACDCERT DELETE.
- To list a certificate, see RACDCERT LIST.

### Syntax

For the key to the symbols used in the command syntax diagrams, see “Syntax of RACF commands and operands” on page 9. The complete syntax of the RACDCERT ADD command is:

```
RACDCERT ADD(data-set-name)
  [ ID(certificate-owner) | SITE | CERTAUTH ]
  [ TRUST | NOTRUST | HIGHTRUST ]
  [ WITHLABEL('label-name') ]
  [ PASSWORD('pkcs12-password') ]
  [ PCICC(['pkds-label' | *]) | ICSF(['pkds-label' | *]) ]
```

If you specify more than one RACDCERT function, only the last specified function is processed. Extraneous keywords that are not related to the function being performed are ignored.

If you do not specify a RACDCERT function, LIST is the default function.

For information on issuing this command as a RACF TSO command, refer to Chapter 3, “RACF TSO commands,” on page 15.

### Parameters

- **ADD(data-set-name)**
  Specifies the data set containing one digital certificate or certificate package. You must specify a cataloged data set, and it may not be a PDS or a PDS member. The record format (RECFM) expected by RACDCERT is variable-byte.
RACDCERT ADD

(VB). When you specify the ADD function, RACDCERT dynamically allocates and opens the specified data set, and reads the certificate from it as binary data.

**ID**(*certificate-owner*) | SITE | CERTAUTH

Specifies that the new certificate is either a user certificate associated with the specified user ID, a site certificate, or a certificate-authority certificate. If you do not specify ID, SITE, or CERTAUTH, the default is ID, and *certificate-owner* defaults to the user ID of the command issuer. If more than one keyword is specified, the last specified keyword is processed and the others are ignored by TSO command parse processing.

**TRUST** | NOTRUST | HIGHTRUST

When specified with the ADD keyword, indicates whether the status of the certificate being added is trusted, not trusted, or highly trusted. Whether a certificate is not trusted or trusted depends on whether or not the certificate is valid and whether the private key has been compromised or not.

Since highly trusted certificates are by definition trusted certificates, any certificate usage that was enabled by marking the certificate trusted will also be enabled by marking the certificate highly trusted. However, only certificate-authority certificates can be highly trusted. The trust status is stored in the UACC field of the certificate profile:

- `X'00'` indicates the status is trusted
- `X'80'` indicates the status is not trusted
- `X'C0'` indicates the status is highly trusted

When a certificate is trusted, it can be used by RACF for its intended purpose (map to a user ID, or treat as a trusted certificate authority or trusted site).

For a personal certificate, TRUST indicates that the certificate can be used to authenticate a user ID.

For a certificate-authority certificate, a trusted certificate is one that can be used to authenticate a user’s certificate by indicating that the entity identified in the certificate (for example, the certificate authority) can issue certificates that this system honors. This implies that a user can gain access to the system based on the information contained in the certificate if the user’s certificate was signed by a trusted certificate authority.

For site certificates, a trusted certificate is one indicating that the entity identified in the certificate (for example, the site) can gain access to the system based on information contained within the certificate. Since the authority that issued the certificate might not be defined to the system as a certificate authority, this certificate information might not be able to be authenticated.

TRUST should only be specified if the command issuer knows:

- This is a valid certificate for this user, site, or certificate authority.
- The private key related to this certificate has not been compromised.

If no trust value is specified on the command, the following processing will take place to determine the trust status:

- If the certificate’s signature can be verified, the certificate has not expired, and the certificate’s validity date range is within the validity date range of the certifying authority’s certificate, the trust status is set to the trust status of the certifying authority’s certificate. For self-signed certificates the certificate being added is set to TRUST by default. If the self-signed certificate has already been added, its trust status is not changed.
If the certificate has expired, has an incorrect validity date range, or cannot be verified because it either has an unknown encryption algorithm or RACF cannot locate its certifying authority’s certificate, the status is set to NOTRUST by default.

If the trust status is to be set from the status of the certifying certificate and the certifying certificate is highly trusted, the status will be trusted.

If the certificate’s signature is incorrect, the certificate is not added.

The TRUST keyword is unrelated to the trusted attribute as defined in the started procedures table (ICHRIN03).

**WITHLABEL(** *label-name*)

Specifies the label to be associated with the certificate. Up to 32 characters can be specified. The *label-name* can contain blanks and mixed-case characters.

This label is used as a handle instead of the serial number and issuer’s distinguished name. It can be used to store a descriptive text.

If the value specified in WITHLABEL already exists, RACDCERT returns a message indicating that the label has already been used. The certificate is not added.

If the user did not specify WITHLABEL, and the data set being processed is PKCS #12, RACF generates the label based on the certificate’s friendly name, which is extracted from the PKCS #12 package and truncated to 32 characters if required.

If WITHLABEL is not specified, or extracted from the PKCS #12 package, RACDCERT generates a label for the certificate. The generated label is of the form LABELnnnnnnnn, where nnnnnnnn is the first integer value, starting at 00000001 that generates a unique label name.

The *label-name* is stripped of leading and trailing blanks. If a single quotation mark is intended to be part of the *label-name*, use two single quotation marks together for each single quotation mark within the string, and enclose the entire string within single quotation marks.

**PASSWORD(** *pkcs12-password*)

Specifies the password that is associated with the PKCS #12 certificate package. This keyword is required if the data set is PKCS #12 and it must not be specified if the data set is not PKCS #12.

**Note:** The password specified will be visible on the screen, so care should be taken to prevent it from being viewed when entered. Because PKCS #12 passwords do not follow the normal TSO/E rules for password content, they cannot be suppressed as they normally would be.

The *pkcs12-password* can be up to 255 characters in length, is case sensitive, and can contain blanks.

**PCICC**([*pkds-label*] | *ICSF*([*pkds-label*] | *))

Specifies that RACF should attempt to store the public or private key associated with this certificate in the ICSF PKA key data set (PKDS). This applies when the key is introduced to RACF by issuing the ADD keyword, and when an existing certificate profile is replaced by issuing the ADD keyword.

The default action for a new key is for RACF to store it as a software key (in the RACF database, not in the ICSF PKDS). The default action for an existing key is to leave it unchanged.
When you specify PCICC with a PKDS label or an asterisk (*):

- If the certificate has a private key, PCICC specifies that the private key is to be converted using a PCI-class cryptographic coprocessor to an ICSF RSA Chinese Remainder Theorem (CRT) key token. The resulting private key is stored in the ICSF PKDS.

- If the certificate has no private key, PCICC specifies that the public key is to be stored as an ICSF RSA Modulus-Exponent (ME) key token.

If the certificate has no private key and you specify PCICC without a PKDS label and without an asterisk (*), the PCICC keyword is ignored. (No PKDS entry is created.)

**PCICC system requirements:**

- When specifying PCICC:
  - The ICSF subsystem must be operational and configured for PKA operations.
  - A PCI-class cryptographic coprocessor must also be operational and configured with this PKDS.

  **Note:** The Crypto Express2 coprocessor feature is required to process keys longer than 2048 bits.

Otherwise, command processing stops and an error message is displayed.

- When the private key is stored as a PCICC key, any system or SSL application using the key must have:
  - An operational ICSF subsystem configured for PKA operations with this PKDS.
  - An operational PCI-class cryptographic coprocessor configured with this PKDS.

**ICSF system requirements:**

Sspecifies that the public or private key is to be converted to an ICSF RSA Modulus-Exponent (ME) key token. The resulting key is stored in the ICSF PKDS.

If the certificate has no private key and you specify ICSF without a PKDS label and without an asterisk (*), the ICSF keyword is ignored. (No PKDS entry is created.)

**ICSF system requirements:**

- When specifying ICSF, the ICSF subsystem must be operational and configured for PKA operations. Otherwise, command processing stops and an error message is displayed.

- When the private key is stored as an ICSF key, any system or SSL application using the key must have ICSF operational and configured for PKA operations with this PKDS.

**Guidelines:**

- Choose the ICSF or PCICC keyword when the private key is RSA, and you want hardware protection for the key.
  - The PCICC key type provides the best performance and supports RSA key sizes up to 4096 bits, but requires a PCI-class cryptographic coprocessor.
The ICSF key type can be used on a PCI-class cryptographic coprocessor or the older cryptographic coprocessor facility. However, the key size is limited to 1024 bits.

Choose neither the ICSF nor PCICC keyword when no hardware protection is desired, or when the private key is DSA.

**PKDS label considerations for both the PCICC and ICSF keywords:**

- The PKDS label for the key is optional. You can specify asterisk (*) to use the PKDS label from the WITHLABEL keyword. If you specify an asterisk (*), you must specify the WITHLABEL keyword.
- Whether specified or taken from the WITHLABEL keyword, the PKDS label must be unique and meet ICSF syntax requirements. That is, allowed characters are alphanumeric, national (ø, ù, §), or period (.). The first character must be alphabetic or national. The label must be 1–64 characters and is translated to uppercase (not case-sensitive).
- For private keys only, when you do not specify a PKDS label and you do not specify an asterisk (*), RACF generates a default label in the format IRR.DIGTCERT.certificate-owner.cvtsname.ebcdic-stck-value, where certificate-owner is the owning user ID, cvtsname is the system name (taken from the CVT), and ebcdic-stck-value is an EBCDIC version of the current store-clock value. RACF does not generate a PKDS label for a public key.

**If the certificate you are adding has an associated private key:** The PCICC or ICSF keyword is ignored and no ICSF PKCS entry is created under any one of these conditions:

- The private key uses the DSA algorithm.
- The private key has already been stored as an ICSF or PCICC key.

**If the certificate you are adding has an existing public or private key in the PKDS, and the key is already associated with this certificate:**

- If the public key already exists in the PKDS and you add its matching private key certificate, you must specify the ICSF or PCICC keyword. (This upgrades the PKDS entry to include the new ICSF or PCICC private key.)
- The PKDS label of the existing public or private key is not changed when you respecify the label with the ICSF or PCICC keyword.
- If the private key already exists in the PKDS as an ICSF key, specifying PCICC will not convert the key to a PCICC key.
- If the private key already exists in the PKDS as a PCICC key, specifying ICSF will not convert the key to an ICSF key.

**If the certificate you are adding has an existing public or private key in the PKDS but the key is not already associated with this (or another) RACF certificate:** RACF associates the key with this certificate when all of the following conditions are true:

- You specify the PKDS label of the existing key using the ICSF or PCICC keyword.
- The existing key is either an RSA encrypted private key (an ICSF internal key token) or an RSA public key.
- No private key is already associated with this certificate.
**RACDCERT ADD**

**Examples**

*Table 24. RACDCERT ADD examples*

**Example 1**

*Operation* User RACFADM with SPECIAL authority requests the addition of a digital certificate for user NETB0Y. RACFADM has placed the digital certificate in data set 'RACFADM.NETB0Y.CERT' and wants the status of the certificate to be trusted.

*Known* User RACFADM has SPECIAL authority. RACFADM has placed the digital certificate in data set 'RACFADM.NETB0Y.CERT'.

*Command* RACDCERT ADD('RACFADM.NETB0Y.CERT')

   ID(NETB0Y)

   TRUST WITHLABEL('Savings Account')

*Output* None.

**Example 2**

*Operation* User WENTING exports her new certificate using the RACDCERT EXPORT command and sends it to her business partner Yun. When he receives it, Yun adds it to his company’s RACF data base as a SITE certificate using the RACDCERT ADD command and calls it Wen Ting.

*Known* The exported certificate does not contain the private key so the data set Wen Ting transmits to Yun need not be protected in any way.

*Commands* Wen Ting's RACDCERT EXPORT command:

   RACDCERT EXPORT(LABEL('Wen Ting's certificate')) DSN(FOR.YUN.CRT)

   Yun's RACDCERT ADD command:

   RACDCERT SITE ADD(WENTING.CRT) WITHLABEL('Wen Ting') ICSF(*)

*Output* None. (To see Yun’s new SITE certificate listed using the RACDCERT LIST command, see [Figure 39 on page 352](#))
RACDCERT ADDRING (Add key ring)

Purpose

Use the RACDCERT ADDRING command to create a new key ring.

Issuing options

The following table identifies the eligible options for issuing the RACDCERT ADDRING command:

<table>
<thead>
<tr>
<th>As a RACF TSO command?</th>
<th>As a RACF operator command?</th>
<th>With command direction?</th>
<th>With automatic command direction?</th>
<th>From the RACF parameter library?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>No</td>
<td>No. (See rules.)</td>
<td>No. (See rules.)</td>
<td>No</td>
</tr>
</tbody>
</table>

Rules:
- The RACDCERT command cannot be directed to a remote system using the AT or ONLYAT keyword.
- The updates made to the RACF database by RACDCERT are eligible for propagation with automatic direction of application updates based on the RRSFDATA profiles AUTODIRECT.target-node.DIGTCERT.APPL and AUTODIRECT.target-node.DIGTRING.APPL, where target-node is the remote node to which the update is to be propagated.

Authorization required

To issue the RACDCERT ADDRING command, you must have the SPECIAL attribute or sufficient authority to the IRR.DIGTCERT.ADDRING resource in the FACILITY class for your intended purpose.

Table 25. Authority required for the RACDCERT ADDRING function

<table>
<thead>
<tr>
<th>IRR.DIGTCERT.ADDRING</th>
<th>Access level</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>READ</td>
<td>READ</td>
<td>Create a key ring for your own user ID.</td>
</tr>
<tr>
<td>UPDATE</td>
<td>UPDATE</td>
<td>Create a key ring for another user.</td>
</tr>
</tbody>
</table>

Activating your changes

If the DIGTRING class is RACLISTed, refresh the class to activate your changes.

Example:

SETROPTS RACLIST(DIGTRING) REFRESH

Related commands

- To delete a key ring, see RACDCERT DELRING.
- To list a key ring, see RACDCERT LISTRING.

Syntax

For the key to the symbols used in the command syntax diagrams, see "Syntax of RACF commands and operands" on page 9. The complete syntax of the RACDCERT ADDRING command is:
RACDCERT ADDRING

RACDCERT ADDRING(\textit{ring-name})

\begin{itemize}
  \item [\text{ID}(\textit{ring-owner})]
\end{itemize}

If you specify more than one RACDCERT function, only the last specified function is processed. Extraneous keywords that are not related to the function being performed are ignored.

If you do not specify a RACDCERT function, LIST is the default function.

For information on issuing this command as a RACF TSO command, refer to Chapter 3, “RACF TSO commands,” on page 15.

Parameters

\textbf{ADDRING(\textit{ring-name})}

Specifies the name of the new key ring. This key ring must not already exist for this user. The new key ring belongs to the user ID specified or defaulted by the \text{ID}(\textit{ring-owner}) keyword.

The key ring name can be up to 237 characters in length. Lowercase characters are permitted. Key ring names become names of RACF profiles in the DIGTRING class, and can contain only characters that are allowed in RACF profile names, with the following restrictions.

\textbf{Restrictions:} The \textit{ring-name} cannot contain any of the following characters:

- an ampersand (X'50')
- an asterisk (X'5C')
- a percent sign (X'6C')

Because only user IDs can have key rings, neither CERTAUTH nor SITE can be specified with ADDRING.

\textbf{ID(\textit{ring-owner})}

Specifies the user ID of the key ring owner. (Only a user ID can have a key ring.) If not specified, the key ring owner defaults to the command issuer's user ID.

Examples

\textbf{Table 26. RACDCERT ADDRING examples}

\textbf{Example 1}

\textbf{Operation}  
User RACFADM wants to add a key ring for the local FTP server. The user ID of the FTP is FTPD. The keys that will be connected to the new ring will be shared by multiple users and the ring will represent the installation's FTP trust policy.

\textbf{Known}  
User RACFADM has SPECIAL authority.

\textbf{Command}  
\texttt{RACDCERT ID(FTPD) ADDRING(FTPring)}

\textbf{Output}  
None.

\textbf{Example 2}

\textbf{Operation}  
User RACFADM wants to add a key ring for a new Web server application. The user ID of the Web server application is WEBSRV02. The keys that will be connected to the new ring will be shared by multiple users and the ring will represent the installation's trust policy for this Web server application.

\textbf{Known}  
User RACFADM has SPECIAL authority.

\textbf{Command}  
\texttt{RACDCERT ADDRING(SSLring) ID(WEBSRV02)}

\textbf{Output}  
None.
RACDCERT ADDTOKEN (Add token)

Purpose

Use the RACDCERT ADDTOKEN command to create a new z/OS PKCS #11 token.

Issuing options

The following table identifies the eligible options for issuing the RACDCERT ADDTOKEN command:

<table>
<thead>
<tr>
<th>As a RACF TSO command?</th>
<th>As a RACF operator command?</th>
<th>With command direction?</th>
<th>With automatic command direction?</th>
<th>From the RACF parameter library?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>No</td>
<td>No. (See rules.)</td>
<td>No. (See rules.)</td>
<td>No</td>
</tr>
</tbody>
</table>

Rules:

- The RACDCERT command cannot be directed to a remote system using the AT or ONLYAT keyword.
- The updates made to the RACF database by RACDCERT are eligible for propagation with automatic direction of application updates based on the RRSFDATA profiles AUTODIRECT.target-node.DIGTCERT.APPL and AUTODIRECT.target-node.DIGTRING.APPL, where target-node is the remote node to which the update is to be propagated.

Authorization required

To issue the RACDCERT ADDTOKEN command, you must have sufficient authority to the appropriate resource in the CRYPTOZ class. (No authority to resources in the FACILITY class is required.) If the command issuer does not have authority to create the specified token as determined by ICSF, the command stops and an error message is displayed.

If the CSFSERV class is active, the command issuer might also need sufficient authority in the CSFSERV class.

For authorization details about the CRYPTOZ and CSFSERV classes, see z/OS Cryptographic Services ICSF Administrator's Guide.

Related commands

- To delete a token, see RACDCERT DELTOKEN.
- To list a token, see RACDCERT LISTTOKEN.

Syntax

For the key to the symbols used in the command syntax diagrams, see Syntax of RACF commands and operands on page 9. The complete syntax of the RACDCERT ADDTOKEN command is:

RACDCERT ADDTOKEN(token-name)

Note: The ID(certificate-owner) | SITE | CERTAUTH parameter is ignored for this RACDCERT function.
RACDCERT ADDTOKEN

If you specify more than one RACDCERT function, only the last specified function is processed. Extraneous keywords that are not related to the function being performed are ignored.

If you do not specify a RACDCERT function, LIST is the default function.

For information on issuing this command as a RACF TSO command, refer to Chapter 3, “RACF TSO commands,” on page 15.

Parameters

ADDTOKEN(token-name)

The token-name value is the name of the token being created. This token must not already exist.

Examples

Table 27. RACDCERT ADDTOKEN examples

Example 1  Operation  User RACFADM wants to create tokens for two servers that have existing RACF certificates.
Known  User RACFADM has SPECIAL authority. The RACF certificate for each server already exists.

Commands  RACDCERT ADDTOKEN(ftperv.ftp.server.pkcs11.token)
RACDCERT ADDTOKEN(websrv.web.server.pkcs11.token)

Output  None.
RACDCERT ALTER (Alter certificate)

Purpose

Use the RACDCERT ALTER command to change the status or the label of a digital certificate for the specified user ID, certificate-authority certificate, or site certificate.

Restriction: Because PKCS #11 tokens are managed by ICSF, not RACF, when you use the RACDCERT ALTER command to alter a certificate that is bound in a token, the change is not reflected on the corresponding certificate object in the token.

See “UTF-8 and BMP character restrictions” on page 278 for information about how UTF-8 and BMP characters in certificate names and labels are processed by RACDCERT functions.

Issuing options

The following table identifies the eligible options for issuing the RACDCERT ALTER command:

<table>
<thead>
<tr>
<th>As a RACF TSO command?</th>
<th>As a RACF operator command?</th>
<th>With command direction?</th>
<th>With automatic command direction?</th>
<th>From the RACF parameter library?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>No</td>
<td>No. (See rules.)</td>
<td>No. (See rules.)</td>
<td>No</td>
</tr>
</tbody>
</table>

Rules:

- The RACDCERT command cannot be directed to a remote system using the AT or ONLYAT keyword.
- The updates made to the RACF database by RACDCERT are eligible for propagation with automatic direction of application updates based on the RRSFCDATA profiles AUTODIRECT.target-node.DIGTCERT.APPL and AUTODIRECT.target-node.DIGTRING.APPL, where target-node is the remote node to which the update is to be propagated.

Authorization required

To issue the RACDCERT ALTER command, you must have the SPECIAL attribute or sufficient authority to the IRR.DIGTCERT.ALTER resource in the FACILITY class for your intended purpose.

Table 28. Authority required for the RACDCERT ALTER function

<table>
<thead>
<tr>
<th>IRR.DIGTCERT.ALTER</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Access level</td>
<td>Purpose</td>
</tr>
<tr>
<td>READ</td>
<td>Change the trust status or label of your own certificate.</td>
</tr>
<tr>
<td>UPDATE</td>
<td>Change the trust status or label of another user’s certificate.</td>
</tr>
<tr>
<td>CONTROL</td>
<td>Change the trust status or label of a SITE or CERTAUTH certificate.</td>
</tr>
</tbody>
</table>

Activating your changes

If the DIGTCERT or DIGTRING class is RACLISTed, refresh the classes to activate your changes.

Example:

```
SETROPTS RACLIST(DIGTCERT, DIGTRING) REFRESH
```
RACDCERT ALTER

Related commands

- To add a certificate, see RACDCERT ADD.
- To delete a certificate, see RACDCERT DELETE.
- To list a certificate, see RACDCERT LIST.

Syntax

For the key to the symbols used in the command syntax diagrams, see “Syntax of RACF commands and operands” on page 9. The complete syntax of the RACDCERT ALTER command is:

```
RACDCERT ALTER
   [ (LABEL('label-name')) ]
   | [ (SERIALNUMBER(serial-number) ISSUERSDN('issuer’s-dn')) ]
   [ ID(certificate-owner) | SITE | CERTAUTH ]
   [ TRUST | NOTRUST | HIGHTRUST ]
   [ NEWLABEL('label-name') ]
```

If you specify more than one RACDCERT function, only the last specified function is processed. Extraneous keywords that are not related to the function being performed are ignored.

If you do not specify a RACDCERT function, LIST is the default function.

For information on issuing this command as a RACF TSO command, refer to Chapter 3, “RACF TSO commands,” on page 15.

Parameters

- `ALTER(LABEL('label-name'))`
- `ALTER(SERIALNUMBER(serial-number) ISSUERSDN('issuer’s-dn'))`
  The TRUST, NOTRUST, or NEWLABEL keyword must be specified with the ALTER keyword. If the user has only one certificate, the SERIALNUMBER and ISSUERSDN keywords, or the LABEL keyword, and their associated values can be omitted. If the user has more than one certificate the LABEL, SERIALNUMBER, or SERIALNUMBER and ISSUERSDN must be used to select which certificate to alter. When specifying the issuer’s distinguished name or the label, the case and blanks displayed when the digital certificate information is listed must be maintained in the ISSUERSDN or the LABEL keyword.
  
  For a description of `label-name`, see the WITHLABEL keyword for RACDCERT ADD.
  
  Note that the only alterable certificate information is the TRUST status or the label of a certificate.

- `ID(certificate-owner) | SITE | CERTAUTH`
  Specifies that the certificate to alter is either a user certificate associated with the specified user ID, a site certificate, or a certificate-authority certificate. If you do not specify ID, SITE, or CERTAUTH, the default is ID, and `certificate-owner` defaults to the user ID of the command issuer. If more than one keyword is specified, the last specified keyword is processed and the others are ignored by TSO command parse processing.

- `TRUST | NOTRUST | HIGHTRUST`
  When specified with the ALTER keyword, indicates whether the status of the
certificate being altered is trusted, not trusted or highly trusted. If TRUST,
NOTRUST, or HIGHTRUST is not specified with the ALTER keyword, no
change to the status of the certificate is attempted.

For a detailed description, see the TRUST, NOTRUST, HIGHTRUST keyword
for RACDCERT ADD.

**NEWLABEL**('new-label-name')

Specifies the label replacing the previous label (if there was one specified) that
is assigned to a certificate.

See the WITHLABEL keyword for RACDCERT ADD for information on label
rules.

If *new-label-name* is the same as *label-name*, the label is not changed and no
message is issued.

**Examples**

*Table 29. RACDCERT ALTER examples*

**Example 1**

<table>
<thead>
<tr>
<th>Operation</th>
<th>User CADUDE with CONTROL access to FACILITY class profile IRR.DIGTCERT.* wishes to mark the local certificate authority highly trusted.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Known</td>
<td>User CADUDE has CONTROL authority to the profile IRR.DIGTCERT.* in the FACILITY class.</td>
</tr>
<tr>
<td>Command</td>
<td>RACDCERT CERTAUTH ALTER(LABEL('Local PKIX CA')) HIGHTRUST</td>
</tr>
<tr>
<td>Output</td>
<td>None.</td>
</tr>
</tbody>
</table>
RACDCERT ALTMAP

RACDCERT ALTMAP (Alter mapping)

Purpose

Use the RACDCERT ALTMAP command to change the label, trust status, or criteria associated with the specified mapping.

See “UTF-8 and BMP character restrictions” on page 278 for information about how UTF-8 and BMP characters in certificate names and labels are processed by RACDCERT functions.

Issuing options

The following table identifies the eligible options for issuing the RACDCERT ALTMAP command:

<table>
<thead>
<tr>
<th>As a RACF TSO command?</th>
<th>As a RACF operator command?</th>
<th>With command direction?</th>
<th>With automatic command direction?</th>
<th>From the RACF parameter library?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>No</td>
<td>No. (See rules.)</td>
<td>No. (See rules.)</td>
<td>No</td>
</tr>
</tbody>
</table>

Rules:

• The RACDCERT command cannot be directed to a remote system using the AT or ONLYAT keyword.
• The updates made to the RACF database by RACDCERT are eligible for propagation with automatic direction of application updates based on the RRSFDATA profiles AUTODIRECT.target-node.DIGTMAP.APPL and AUTODIRECT.target-node.DIGTCRIT.APPL, where target-node is the remote node to which the update is to be propagated.

Authorization required

To issue the RACDCERT ALTMAP command, you must have the SPECIAL attribute or sufficient authority to the IRR.DIGTCERT.ALTMAP resource in the FACILITY class for your intended purpose.

Table 30. Authority required for the RACDCERT ALTMAP function

<table>
<thead>
<tr>
<th>IRR.DIGTCERT.ALTMAP</th>
<th>Access level</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>READ</td>
<td></td>
<td>Alter a mapping associated with your own user ID.</td>
</tr>
<tr>
<td>UPDATE</td>
<td></td>
<td>Alter a mapping associated with another user ID or MULTIID.</td>
</tr>
</tbody>
</table>

Activating your changes

If the DIGTNMAP or DIGTCRIT class is RACLISTed, refresh the classes to activate your changes.

Example:

SETROPTS RACLIST(DIGTNMAP, DIGTCRIT) REFRESH

Related commands

• To define a user ID mapping, see RACDCERT MAP.
• To delete a user ID mapping, see RACDCERT DELMAP.
• To list a user ID mapping, see RACDCERT LISTMAP.
RACDCERT ALTMAP

Syntax

For the key to the symbols used in the command syntax diagrams, see "Syntax of RACF commands and operands" on page 9. The complete syntax of the RACDCERT ALTMAP command is:

```
RACDCERT ALTMAP[(LABEL('label-name'))]
   [ ID(mapping-owner) | MULTIID ]
   [ NEWCRITERIA(criteria-profile-name-template) ]
   [ NEWLABEL('label-name') ]
   [ TRUST | NOTRUST ]
```

If you specify more than one RACDCERT function, only the last specified function is processed. Extraneous keywords that are not related to the function being performed are ignored.

If you do not specify a RACDCERT function, LIST is the default function.

For information on issuing this command as a RACF TSO command, refer to Chapter 3, “RACF TSO commands,” on page 15.

Parameters

ALTMAP
ALTMAP(LABEL('label-name'))
   Specifying label name is required if more than one mapping is associated with the user ID. If NEWLABEL, NEWCRITERIA, or TRUST/NOTRUST is not specified, the mapping is not altered.

ID(mapping-owner) | MULTIID
   Specifies the user ID associated with the mapping. If you do not specify ID or MULTIID, the default is ID, and mapping-owner defaults to the user ID of the command issuer. If more than one keyword is specified, the last specified keyword is processed and the others are ignored by TSO command parse processing.

   ID(mapping-owner)
   Specifies the user ID associated with the mapping.

MULTIID
   Specifies that additional criteria is used to determine the user ID associated with the mapping.

NEWCRITERIA(criteria-profile-name-template)
   Changes the template associated with this mapping when specified with MULTIID. New DIGTCRIT profiles must be created to match the new template profile names. NEWCRITERIA can only be set for MULTIID.

NEWLABEL('new-label-name')
   Specifies the label replacing the previous label assigned to a certificate mapping.

   See the WITHLABEL keyword for RACDCERT ADD for information on label rules.

   If new-label-name is the same as label-name, the label is not changed and no message is issued.
RACDCERT ALTMAP

TRUST I NOTRUST

When specified with ALTMAP, indicates whether this mapping can be used to associate a user ID to a certificate presented by a user accessing the system.

Examples

Table 31. RACDCERT ALTMAP examples

Example 1

Operation  User RACFADM with SPECIAL authority has been notified by the network administrator that the users in department BWVA can begin using their certificates to access the system. The mapping previously created with the label BWVA USERS can now be marked trusted.

Known  User RACFADM has SPECIAL authority.

Command  RACDCERT ID(BWVAUSR) ALTMAP(LABEL('BWVA USERS')) TRUST

Output  None.
RACDCERT BIND (Bind certificate to token)

Purpose

Use the RACDCERT BIND command to bind a digital certificate to a z/OS PKCS #11 token.

**Rule:** The certificate must be added to the RACF database by a RACDCERT ADD or RACDCERT GENCERT command *prior* to issuing the RACDCERT BIND command.

When a certificate is bound to a token, RACF creates the following objects in the token:
- a certificate object
- a public key object
- a private key object, if the certificate has an associated private key and the BIND USAGE is PERSONAL.

**Unsupported key types:** Command processing stops and an error message is displayed if the target certificate uses an unsupported key type.
- DSA public keys are not supported.
- DSA, ICSF, and PCICC private keys are not supported. (The target private key of the bind operation must be a non-ICSF key.)

See [UTF-8 and BMP character restrictions on page 278](#) for information about how UTF-8 and BMP characters in certificate names and labels are processed by RACDCERT functions.

Issuing options

The following table identifies the eligible options for issuing the RACDCERT BIND command:

<table>
<thead>
<tr>
<th>As a RACF TSO command?</th>
<th>As a RACF operator command?</th>
<th>With command direction?</th>
<th>With automatic command direction?</th>
<th>From the RACF parameter library?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>No</td>
<td>No. (See rules.)</td>
<td>No. (See rules.)</td>
<td>No</td>
</tr>
</tbody>
</table>

**Rules:**
- The RACDCERT command cannot be directed to a remote system using the AT or ONLYAT keyword.
- The updates made to the RACF database by RACDCERT are eligible for propagation with automatic direction of application updates based on the RRSFCDATA profiles AUTODIRECT.target-node.DIGTCERT.APPL and AUTODIRECT.target-node.DIGTRING.APPL, where target-node is the remote node to which the update is to be propagated.

Authorization required

To issue the RACDCERT BIND command, you must have the following authorizations:
- The SPECIAL attribute, or sufficient authority to the IRR.DIGTCERT.BIND resource in the FACILITY class based on the USAGE value.
- The SPECIAL attribute, or sufficient authority to the IRR.DIGTCERT.ADD resource in the FACILITY class based on the certificate owner.
- Sufficient authority to the appropriate resources in the CRYPTOZ class.
If the CSFSERV class is active, sufficient authority to appropriate resources in the CSFSERV class.

For details about CRYPTOZ and CSFSERV authorities, see z/OS Cryptographic Services ICSF Administrator's Guide.

If the command issuer is not authorized by ICSF (through the CRYPTOZ class) to add the object to the specified token or is not authorized by RACF (through the FACILITY class) to reference the specified RACF certificate, the command stops and an error message is displayed.

Table 32. Authority required for the RACDCERT BIND function

<table>
<thead>
<tr>
<th>USAGE value</th>
<th>Your own certificate</th>
<th>Another user's certificate</th>
<th>SITE or CERTAUTH certificate</th>
</tr>
</thead>
<tbody>
<tr>
<td>PERSONAL</td>
<td>Sufficient authority to CRYPTOZ resources and READ authority to IRR.DIGTCERT.BIND</td>
<td>Sufficient authority to CRYPTOZ resources and UPDATE authority to IRR.DIGTCERT.BIND</td>
<td>Sufficient authority to CRYPTOZ resources and CONTROL authority to IRR.DIGTCERT.BIND</td>
</tr>
<tr>
<td>SITE</td>
<td>Sufficient authority to CRYPTOZ resources, CONTROL authority to IRR.DIGTCERT.ADD and READ authority to IRR.DIGTCERT.BIND</td>
<td>Sufficient authority to CRYPTOZ resources, CONTROL authority to IRR.DIGTCERT.ADD and UPDATE authority to IRR.DIGTCERT.BIND</td>
<td>Sufficient authority to CRYPTOZ resources and UPDATE authority to IRR.DIGTCERT.BIND</td>
</tr>
<tr>
<td>CERTAUTH</td>
<td>Sufficient authority to CRYPTOZ resources, CONTROL authority to IRR.DIGTCERT.ADD and READ authority to IRR.DIGTCERT.BIND</td>
<td>Sufficient authority to CRYPTOZ resources, CONTROL authority to IRR.DIGTCERT.ADD and UPDATE authority to IRR.DIGTCERT.BIND</td>
<td>Sufficient authority to CRYPTOZ resources and UPDATE authority to IRR.DIGTCERT.BIND</td>
</tr>
</tbody>
</table>

Related commands
- To unbind a certificate from a token, see RACDCERT UNBIND.
- To add a token, see RACDCERT ADDTOKEN.
- To list a token, see RACDCERT LISTTOKEN.

Syntax

For the key to the symbols used in the command syntax diagrams, see Syntax of RACF commands and operands on page 9. The complete syntax of the RACDCERT BIND command is:

```
RACDCERT BIND(TOKEN(token-name))
  LABEL('label-name')
  [ID(certificate-owner) | SITE | CERTAUTH]
  [DEFAULT]
  [USAGE(PERSONAL | SITE | CERTAUTH)]
)
```

Note: Unless specified as a subkeyword of the BIND parameter, the ID(certificate-owner) | SITE | CERTAUTH parameter is ignored for the RACDCERT BIND function.

If you specify more than one RACDCERT function, only the last specified function is processed. Extraneous keywords that are not related to the function being performed are ignored.

If you do not specify a RACDCERT function, LIST is the default function.
RACDCERT BIND

For information on issuing this command as a RACF TSO command, refer to Chapter 3, “RACF TSO commands,” on page 15.

Parameters

BIND(TOKEN(token-name) ID(certificate-owner) LABEL('label-name'))
BIND(TOKEN(token-name) SITE LABEL('label-name'))
BIND(TOKEN(token-name) CERTAUTH LABEL('label-name'))

You must uniquely identify both the token and the certificate to be bound.

TOKEN(token-name)
Specifies the name of the token to which the certificate is to be bound. For RACDCERT BIND, you must specify the TOKEN operand.

ID(certificate-owner) | SITE | CERTAUTH
Specifies that the certificate being bound to the token is either a user certificate associated with the specified user ID, a site certificate, or a certificate-authority certificate. If the ID, SITE, and CERTAUTH keywords are omitted, the default is ID, and certificate-owner defaults to the user ID of the command issuer. If more than one keyword is specified, the last specified keyword is processed and the others are ignored by TSO command parse processing.

LABEL('label-name')
Specifies the certificate being bound to the token. For RACDCERT BIND, you must specify the LABEL operand.

DEFAULT
Specifies that the certificate is the default certificate for the token. Only one certificate within the token is the default certificate. If a default certificate already exists, its DEFAULT status is removed, and the specified certificate becomes the default certificate.

You must specify the DEFAULT keyword when you want the specified certificate to be the default certificate for this token.

To remove the DEFAULT status of a certificate without defining another certificate as the default certificate, reissue the RACDCERT BIND command for the certificate and omit the DEFAULT keyword.

USAGE(PERSONAL | SITE | CERTAUTH)
Specifies how this certificate is used within the specified token. If no usage is specified, it defaults to the usage of the certificate being bound.

Specify the USAGE keyword to alter the trust policy of the certificate for a specific token. For example, if you operate your own certificate authority, your certificate server application has its own certificate. Because the certificate represents your certificate authority, it is defined as CERTAUTH, which sets its default usage for all applications and users. Your certificate server application requires use of the certificate’s private key for signing purposes but the default usage of CERTAUTH does not allow this use. To allow it, specify USAGE(PERSONAL) when you bind this certificate to the token of the certificate server application. This allows you to alter the trust policy for this token only without affecting the default usage for all other applications and users.

Important: Carefully control use of the USAGE keyword. RACDCERT processing ensures that the command issuer is authorized to define SITE or CERTAUTH certificates and cannot bypass the installation security policy using the USAGE keyword.
See Table 32 on page 304 for authorization requirements for users without the SPECIAL attribute to allow them to bind a certificate to a token with the PERSONAL, SITE, or CERTAUTH usage.

Examples

Table 33. RACDCERT BIND examples

Example 1

Operation User SECADM wants to bind an existing root CA certificate to two existing tokens.

Known User SECADM has the SPECIAL attribute.

The root CA certificate is installed under CERTAUTH with the label Local Root CA for Servers.

The following tokens are already defined to RACF:

- ftpsrv.ftp.server.pkcs11.token
- websrv.web.server.pkcs11.token

Commands

RACDCERT BIND(CERTAUTH LABEL('Local Root CA for Servers')
TOKEN(ftpsrv.ftp.server.pkcs11.token)
RACDCERT BIND(CERTAUTH LABEL('Local Root CA for Servers')
TOKEN(websrv.web.server.pkcs11.token))

Output None.

Example 2

Operation User SECADM wants to bind end-entity certificates to their respective tokens and define each certificate as the default in its token.

Known User SECADM has the SPECIAL attribute.

The following tokens are already defined to RACF:

- ftpsrv.ftp.server.pkcs11.token
- websrv.web.server.pkcs11.token

An end-entity certificate and private key labeled FTP key is already defined to RACF and installed under the user ID FTPSRV.

An end-entity certificate and private key labeled Web key is already defined to RACF and installed under the user ID WEBSRV.

Both end-entity certificates are signed by the existing root CA certificate labeled Local Root CA for Servers.

Commands

RACDCERT BIND(ID(FTPSRV) LABEL('FTP key') DEFAULT
TOKEN(ftpsrv.ftp.server.pkcs11.token)
RACDCERT BIND(ID(WEBSRV) LABEL('Web key') DEFAULT
TOKEN(websrv.web.server.pkcs11.token))

Output None.
RACDCERT CHECKCERT (Check certificate)

Purpose

Use the RACDCERT CHECKCERT command to check if the digital certificate contained in the specified data set has already been added to the RACF database and associated with a user ID.

See "UTF-8 and BMP character restrictions" on page 278 for information about how UTF-8 and BMP characters in certificate names are displayed using RACDCERT functions.

Issuing options

The following table identifies the eligible options for issuing the RACDCERT CHECKCERT command:

<table>
<thead>
<tr>
<th>As a RACF TSO command?</th>
<th>As a RACF operator command?</th>
<th>With command direction?</th>
<th>With automatic command direction?</th>
<th>From the RACF parameter library?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>No</td>
<td>No. (See rules.)</td>
<td>No. (See rules.)</td>
<td>No</td>
</tr>
</tbody>
</table>

Rules:

- The RACDCERT command cannot be directed to a remote system using the AT or ONLYAT keyword.
- The updates made to the RACF database by RACDCERT are eligible for propagation with automatic direction of application updates based on the RRSFCDATA profiles AUTODIRECT.target-node.DIGTCERT.APPL and AUTODIRECT.target-node.DIGTRING.APPL, where target-node is the remote node to which the update is to be propagated.

Authorization required

To issue the RACDCERT CHECKCERT command, you must have the SPECIAL attribute or sufficient authority to the IRR.DIGTCERT.LIST resource in the FACILITY class for your intended purpose.

Table 34. Authority required for the RACDCERT CHECKCERT function

<table>
<thead>
<tr>
<th>IRR.DIGTCERT.LIST</th>
<th>Access level</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>IRR.DIGTCERT.LIST</td>
<td>READ</td>
<td>Check your own certificate.</td>
</tr>
<tr>
<td></td>
<td>UPDATE</td>
<td>Check another user’s certificate.</td>
</tr>
<tr>
<td></td>
<td>CONTROL</td>
<td>Check a SITE or CERTAUTH certificate.</td>
</tr>
</tbody>
</table>

Related commands

- To add a certificate, see RACDCERT ADD.
- To list a certificate, see RACDCERT LIST.

Syntax

For the key to the symbols used in the command syntax diagrams, see "Syntax of RACF commands and operands" on page 9. The complete syntax of the RACDCERT CHECKCERT command is:

RACDCERT CHECKCERT
RACDCERT CHECKCERT

RACDCERT CHECKCERT(data-set-name)
   [PASSWORD('pkcs12-password')]

Note: The ID(certificate-owner) | SITE | CERTAUTH parameter is ignored for this RACDCERT function.

If you specify more than one RACDCERT function, only the last specified function is processed. Extraneous keywords that are not related to the function being performed are ignored.

If you do not specify a RACDCERT function, LIST is the default function.

For information on issuing this command as a RACF TSO command, refer to Chapter 3, “RACF TSO commands,” on page 15.

Parameters

CHECKCERT(data-set-name)

CHECKCERT lists the certificate in the specified data set. If the certificate request is made by a user with proper authority, information in the RACF database pertaining to that certificate is also displayed. Additionally, an authority check is performed by data management when the data set is opened.

The CHECKCERT keyword also supports the evaluation of site certificates and certificate authority certificates. It indicates if the certificate is defined and to whom it is defined after checking the resource IRR.DIGTCERT.LIST in the FACILITY class. READ authority is required if the certificate is associated with the user issuing the command. UPDATE authority is required if the certificate is associated with a user other than the issuer of the command. CONTROL authority is required if the certificate is a certificate authority or a site certificate.

The CHECKCERT keyword can be used on the same set of certificate packages that is allowed by RACDCERT ADD. See RACDCERT ADD for more information.

Notes:
1. The issuer of the RACDCERT command must have READ access to the data-set-name data set to prevent an authorization abend from occurring when the data set is read.
2. No certificate ID is displayed if the certificate is not installed. If the certificate is installed, the certificate ID is displayed only if the certificate has a label and the user is authorized to list the specific certificate information.

PASSWORD('pkcs12-password')

Specifies the password that is associated with the PKCS #12 certificate package. It is required if the data set contains a PKCS #12 certificate package and it must not be specified if the data set contents are not PKCS #12.

Note: The password specified will be visible on the screen, so care should be taken to prevent it from being viewed when entered. Because PKCS #12 passwords do not follow the normal TSO/E rules for password content, they cannot be suppressed as they normally would be.

The 'pkcs12-password' can be up to 255 characters in length, is case sensitive, and can contain blanks.
Examples

Table 35. RACDCERT CHECKCERT examples

Example 1

Operation User NETADMN has a digital certificate in a data set, and is uncertain who it belongs to, and whether or not it has been defined. The digital certificate is in data set 'NETADMN.SOMEONZ.CERT'. NETADMN has UPDATE authority to the FACILITY class profile IRR.DIGTCERT.LIST. He issues the following RACDCERT, and the output he receives indicates that it has already been defined for user GTM.

Known User NETADMN has been given UPDATE access to profile IRR.DIGTCERT.LIST in the FACILITY class. NETADMN has placed the digital certificate in data set 'NETADMN.SOMEONZ.CERT'.

Command RACDCERT CHECKCERT('NETADMN.SOMEONZ.CERT')

Output See [Figure 37]

```
RACDCERT CHECKCERT('NETADMN.SOMEONZ.CERT')

Serial Number:
>79<
Issuer's Name:
>CN=BobsBank Class 2<
Subject's Name:
>brchMGR@BobsBank.com,CN=J. Miles,T=Manager,OU=Branch2,O=BobsBank,INC.<
>.SP=NY,L=Internet,C=USA<
```

Figure 37. Sample output from the RACDCERT CHECKCERT command
RACDCERT CONNECT

RACDCERT CONNECT (Connect a certificate to key ring)

Purpose

Use the RACDCERT CONNECT command to add a digital certificate to a key ring.

See "UTF-8 and BMP character restrictions" on page 278 for information about how UTF-8 and BMP characters in certificate names and labels are processed by RACDCERT functions.

Issuing options

The following table identifies the eligible options for issuing the RACDCERT CONNECT command:

<table>
<thead>
<tr>
<th>As a RACF TSO command?</th>
<th>As a RACF operator command?</th>
<th>With command direction?</th>
<th>With automatic command direction?</th>
<th>From the RACF parameter library?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>No</td>
<td>No. (See rules.)</td>
<td>No. (See rules.)</td>
<td>No</td>
</tr>
</tbody>
</table>

Rules:

- The RACDCERT command cannot be directed to a remote system using the AT or ONLYAT keyword.
- The updates made to the RACF database by RACDCERT are eligible for propagation with automatic direction of application updates based on the RRSFDATA profiles AUTODIRECT, target-node.DIGTCERT.APPL and AUTODIRECT, target-node.DIGTRING.APPL, where target-node is the remote node to which the update is to be propagated.

Authorization required

To issue the RACDCERT CONNECT command, you must have the SPECIAL attribute or sufficient authority to the following resources in the FACILITY class, based on the certificate owner, key ring owner, and the USAGE value:

- IRR.DIGTCERT.CONNECT
- IRR.DIGTCERT.ADD

The USAGE keyword allows a certificate to be connected to a ring and used in a manner that differs from the certificate's original use. For example, by changing the USAGE value, a certificate defined as a user certificate might be used as a certificate-authority certificate.

The USAGE keyword is powerful and must be controlled. The rules for connection are shown in Table 36 on page 311, which shows the access control checks that are performed when connecting to your own key ring, and Table 37 on page 311, which shows the access control checks that are performed when connecting to another user's key ring.
Table 36. Authority required for the RACDCERT CONNECT function—Connecting to your own key ring

<table>
<thead>
<tr>
<th>USAGE value</th>
<th>Your own certificate</th>
<th>Another user’s certificate</th>
<th>SITE or CERTAUTH certificate</th>
</tr>
</thead>
<tbody>
<tr>
<td>PERSONAL</td>
<td>READ authority to IRR.DIGTCERT.CONNECT</td>
<td>UPDATE authority to IRR.DIGTCERT.CONNECT</td>
<td>CONTROL authority to IRR.DIGTCERT.CONNECT</td>
</tr>
<tr>
<td>SITE</td>
<td>CONTROL authority to IRR.DIGTCERT.CONNECT</td>
<td>CONTROL authority to IRR.DIGTCERT.CONNECT</td>
<td>UPDATE authority to IRR.DIGTCERT.CONNECT</td>
</tr>
<tr>
<td>CERTAUTH</td>
<td>CONTROL authority to IRR.DIGTCERT.CONNECT</td>
<td>CONTROL authority to IRR.DIGTCERT.CONNECT</td>
<td>CONTROL authority to IRR.DIGTCERT.CONNECT</td>
</tr>
</tbody>
</table>

Table 37. Authority required for the RACDCERT CONNECT function—Connecting to another user’s key ring

<table>
<thead>
<tr>
<th>USAGE value</th>
<th>Your own certificate</th>
<th>Another user’s certificate</th>
<th>SITE or CERTAUTH certificate</th>
</tr>
</thead>
<tbody>
<tr>
<td>PERSONAL</td>
<td>CONTROL authority to IRR.DIGTCERT.CONNECT</td>
<td>CONTROL authority to IRR.DIGTCERT.CONNECT</td>
<td>CONTROL authority to IRR.DIGTCERT.CONNECT</td>
</tr>
<tr>
<td>SITE</td>
<td>CONTROL authority to IRR.DIGTCERT.CONNECT</td>
<td>CONTROL authority to IRR.DIGTCERT.CONNECT</td>
<td>CONTROL authority to IRR.DIGTCERT.CONNECT</td>
</tr>
<tr>
<td>CERTAUTH</td>
<td>CONTROL authority to IRR.DIGTCERT.CONNECT</td>
<td>CONTROL authority to IRR.DIGTCERT.CONNECT</td>
<td>CONTROL authority to IRR.DIGTCERT.CONNECT</td>
</tr>
</tbody>
</table>

See the USAGE subkeyword below for additional information on the authority required to change a certificate’s usage.

**Activating your changes**

If the DIGTCERT or DIGTRING class is RACLISTed, refresh the classes to activate your changes.

**Example:**

SETROPTS RACLIST(DIGTCERT, DIGTRING) REFRESH

**Related commands**

- To add a key ring, see RACDCERT ADDRING.
- To remove a certificate from a key ring, see RACDCERT REMOVE.
- To list a key ring, see RACDCERT LISTRING.

**Syntax**

For the key to the symbols used in the command syntax diagrams, see "Syntax of RACF commands and operands" on page 9. The complete syntax of the RACDCERT CONNECT command is:

```
RACDCERT CONNECT([ID(certificate-owner) | SITE | CERTAUTH]
    LABEL('label-name')
    RING('ring-name')
    [DEFAULT]
    [USAGE(PERSONAL | SITE | CERTAUTH)]
    [ID(ring-owner)]
```

If you specify more than one RACDCERT function, only the last specified function is processed. Extraneous keywords that are not related to the function being performed are ignored.
RACDCERT CONNECT

If you do not specify a RACDCERT function, LIST is the default function.

For information on issuing this command as a RACF TSO command, refer to Chapter 3, “RACF TSO commands,” on page 15.

Parameters

CONNECT(ID(certificate-owner) LABEL('label-name') RING(ring-name))
CONNECT(SITE LABEL('label-name') RING(ring-name))
CONNECT(CERTAUTH LABEL('label-name') RING(ring-name))

Specifies the digital certificate to be added to the key ring. The specified certificate must be added to the RACF database by a RACDCERT ADD or RACDCERT GENCERT command prior to issuing the CONNECT command.

ID(certificate-owner) indicates that the certificate being connected is a user certificate, and certificate-owner is the user ID associated with this certificate.
SITE indicates that the certificate being connected is a site certificate, and CERTAUTH indicates that it is a certificate authority certificate. If ID, SITE or CERTAUTH are not specified, ID(certificate-owner) defaults to the key ring owner as specified or defaulted by the ID(ring-owner) keyword.

LABEL('label-name')

Specifies the certificate that is being connected to the key ring. You must specify a label.

RING(ring-name)

Specifies the key ring to which this certificate is being connected. You must specify a ring name. Note: The key ring belongs to the ID specified or defaulted by the ID(ring-owner) keyword.

ID(ring-owner)

Specifies the user ID of the key ring owner. (Only a user ID can have a key ring.) If not specified, the key ring owner defaults to the command issuer’s user ID.

DEFAULT

Specifies that the certificate is the default certificate for the ring. Only one certificate within the key ring can be the default certificate. If a default certificate already exists, its DEFAULT status is removed, and the specified certificate becomes the default certificate. If you want the specified certificate to be the default, DEFAULT must be explicitly specified.

If you have a key ring with a default certificate and you want to remove the default status of the certificate without defining another certificate as the default certificate, CONNECT the certificate again without specifying the DEFAULT keyword.

USAGE(PERSONAL | SITE | CERTAUTH)

Specifies how this certificate is used within the specified ring. If no usage is specified, it defaults to the usage of the certificate being connected.

The USAGE keyword allows the altering of the trust policy within the confines of a specific key ring. For example, if you are operating your own certificate authority, your certificate server application would have its own certificate. Since the certificate does represent a certificate authority, it should be installed under CERTAUTH, thus setting its default usage for all other applications and users. However, your certificate server application would need to use the certificate’s private key for signing. The default usage of CERTAUTH does not allow this. So, for the certificate server application’s key ring only, the certificate should be connected with USAGE(PERSONAL). Note, in addition to the above, the user
ID assigned to your certificate server application needs to be granted permission to operate as a certificate authority. This is done by giving the user ID CONTROL access to FACILITY class resource IRR.DIGTCERT.GENCERT.

For the sake of consistency, other certificate and USAGE variations are supported. However, there is currently no practical application for them.

When using the USAGE keyword to change the usage of a certificate, such as is done when a PERSONAL certificate is being used as a SITE or CERTAUTH certificate, RACDCERT must ensure that you have the ability to define a SITE or CERTAUTH certificate by authenticating that the command issuer has CONTROL authority to the resource IRR.DIGTCERT.ADD in the FACILITY class. This ensures that a user cannot bypass the installation security policy through the use of USAGE.

**Examples**

**Table 38. RACDCERT CONNECT examples**

<table>
<thead>
<tr>
<th>Example 1</th>
<th>Operation</th>
<th>User RACFADM wants to connect an existing SITE certificate labeled Shared Server to the RING01 key ring of server INVSERV. The certificate will be added to the key ring as the default certificate.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Known</td>
<td>User RACFADM has SPECIAL authority.</td>
<td></td>
</tr>
<tr>
<td>Command</td>
<td>RACDCERT ID(INVSERV) CONNECT(SITE LABEL('Shared Server') RING(RING01) USAGE(PERSONAL) DEFAULT)</td>
<td></td>
</tr>
<tr>
<td>Output</td>
<td>None.</td>
<td></td>
</tr>
</tbody>
</table>
RACDCERT DELETE

RACDCERT DELETE (Delete certificate)

Purpose

Use the RACDCERT DELETE command to delete a digital certificate.

When you delete a certificate that is connected to a key ring, the certificate is automatically removed from the key ring.

Restriction: Because PKCS #11 tokens are managed by ICSF, not RACF, when you use the RACDCERT DELETE command to delete a certificate that is bound in a token, the corresponding certificate object remains in the token.

The DELETE function also supports site and certificate-authority certificates, and the deletion of the private key and other certificate data that is stored when the certificate was created.

When a user profile is deleted with the DELUSER command, related DIGTCERT, DIGTRING, and DIGTNMAP profiles are deleted as a part of DELUSER processing. However, under some circumstances, residual profiles might not be deleted. For example, if you issue the DELUSER command from a z/VM® system (which does not support digital certificates), the profiles might not be deleted. The DELETE, DELRING and DELMAP keywords for RACDCERT support the specification of a user ID in order to allow residual certificate information related to the user ID to be deleted. However, the other RACDCERT functions require the user ID to be defined to RACF.

See "UTF-8 and BMP character restrictions" on page 278 for information about how UTF-8 and BMP characters in certificate names and labels are processed by RACDCERT functions.

Issuing options

The following table identifies the eligible options for issuing the RACDCERT DELETE command:

<table>
<thead>
<tr>
<th>As a RACF TSO command?</th>
<th>As a RACF operator command?</th>
<th>With command direction?</th>
<th>With automatic command direction?</th>
<th>From the RACF parameter library?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>No</td>
<td>No. (See rules.)</td>
<td>No. (See rules.)</td>
<td>No</td>
</tr>
</tbody>
</table>

Rules:

- The RACDCERT command cannot be directed to a remote system using the AT or ONLYAT keyword.
- The updates made to the RACF database by RACDCERT are eligible for propagation with automatic direction of application updates based on the RRSFDATA profiles AUTODIRECT.target-node.DIGTCERT.APPL and AUTODIRECT.target-node.DIGTRING.APPL, where target-node is the remote node to which the update is to be propagated.

Authorization required

To issue the RACDCERT DELETE command, you must have the SPECIAL attribute or sufficient authority to the IRR.DIGTCERT.DELETE resource in the FACILITY class for your intended purpose.
RACDCERT DELETE

Table 39. Authority required for the RACDCERT DELETE function

<table>
<thead>
<tr>
<th>Access level</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>READ</td>
<td>Delete your own certificate.</td>
</tr>
<tr>
<td>UPDATE</td>
<td>Delete another user's certificate.</td>
</tr>
<tr>
<td>CONTROL</td>
<td>Delete a SITE or CERTAUTH certificate.</td>
</tr>
</tbody>
</table>

Activating your changes

If the DIGTCERT or DIGTRING class is RA.ToListed, refresh the classes to activate your changes.

Example:

SETROPTS RACLST(DIGTCERT, DIGTRING) REFRESH

Related commands

- To add a certificate, see RACDCERT ADD.
- To alter a certificate, see RACDCERT ALTER.
- To list a certificate, see RACDCERT LIST.

Syntax

For the key to the symbols used in the command syntax diagrams, see “Syntax of RACF commands and operands” on page 9. The complete syntax of the RACDCERT DELETE command is:

```
RACDCERT DELETE
  [ (LABEL('label-name')) ]
  | [ (SERIALNUMBER(serial-number) [ ISSUERSDN('issuer's-dn') ] ) ]
  [ ID(certificate-owner) | SITE | CERTAUTH ]
```

If you specify more than one RACDCERT function, only the last specified function is processed. Extraneous keywords that are not related to the function being performed are ignored.

If you do not specify a RACDCERT function, LIST is the default function.

For information on issuing this command as a RACF TSO command, refer to Chapter 3, “RACF TSO commands,” on page 15.

Parameters

**DELETE(LABEL('label-name'))**

**DELETE(SERIALNUMBER(serial-number) ISSUERSDN('issuer's-dn'))**

If the user has only one certificate, the SERIALNUMBER and ISSUERSDN keywords, or the LABEL keyword, and their associated values can be omitted. If the user has more than one certificate the LABEL, SERIALNUMBER, or SERIALNUMBER and ISSUERSDN must be used to select which certificate to delete. When specifying the issuer's distinguished name or the label, the mixed-case characters and blanks displayed when the digital certificate information is listed must be maintained in the ISSUERSDN or the LABEL keyword.
RACDCERT DELETE

For a description of label-name, see the description of the WITHLABEL keyword for RACDCERT ADD.

**ID(certificate-owner) | SITE | CERTAUTH**

Specifies that the specified certificate is either a user certificate associated with the specified user ID, a site certificate, or a certificate-authority certificate. If you do not specify ID, SITE, or CERTAUTH, the default is ID, and certificate-owner defaults to the user ID of the command issuer. If more than one keyword is specified, the last specified keyword is processed and the others are ignored by TSO command parse processing.

**Examples**

Table 40. RACDCERT DELETE examples

**Example 1**

*Operation* User RACFADM wants to delete the digital certificate labeled Savings Account for user NETB0Y.

*Known* User RACFADM has SPECIAL authority.

*Command* RACDCERT DELETE(LABEL('Savings Account')) ID(NETB0Y)

*Output* None.
RACDCERT DELMAP (Delete mapping)

Purpose

Use the RACDCERT DELMAP command to delete a mapping for a user ID.

See “UTF-8 and BMP character restrictions” on page 278 for information about how UTF-8 and BMP characters in certificate names and labels are processed by RACDCERT functions.

Issuing options

The following table identifies the eligible options for issuing the RACDCERT DELMAP command:

<table>
<thead>
<tr>
<th>As a RACF TSO command?</th>
<th>As a RACF operator command?</th>
<th>With command direction?</th>
<th>With automatic command direction?</th>
<th>From the RACF parameter library?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>No</td>
<td>No. (See rules.)</td>
<td>No. (See rules.)</td>
<td>No</td>
</tr>
</tbody>
</table>

Rules:

- The RACDCERT command cannot be directed to a remote system using the AT or ONLYAT keyword.
- The updates made to the RACF database by RACDCERT are eligible for propagation with automatic direction of application updates based on the RRSFDATA profiles AUTODIRECT.target-node.DIGTMAP.APPL and AUTODIRECT.target-node.DIGTCRIT.APPL, where target-node is the remote node to which the update is to be propagated.

Authorization required

To issue the RACDCERT DELMAP command, you must have the SPECIAL attribute or sufficient authority to the IRR.DIGTCERT.DELMAP resource in the FACILITY class for your intended purpose.

Table 41. Authority required for the RACDCERT DELMAP function

<table>
<thead>
<tr>
<th>IRR.DIGTCERT.DELMAP</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Access level</td>
<td>Purpose</td>
</tr>
<tr>
<td>READ</td>
<td>Delete a mapping associated with your own user ID.</td>
</tr>
<tr>
<td>UPDATE</td>
<td>Delete a mapping associated with another user ID or MULTIID.</td>
</tr>
</tbody>
</table>

Activating your changes

If the DIGTMAP or DIGTCRIT class is RACLISTed, refresh the classes to activate your changes.

Example:

SETROPTS RACLIST(DIGTMAP, DIGTCRIT) REFRESH

Related commands

- To define a user ID mapping, see RACDCERT MAP.
- To alter a user ID mapping, see RACDCERT ALTMAP.
- To list a user ID mapping, see RACDCERT LISTMAP.
The RACDCERT DELMAP command is unrelated to the RACMAP DELMAP command.

Syntax

For the key to the symbols used in the command syntax diagrams, see “Syntax of RACF commands and operands” on page 9. The complete syntax of the RACDCERT DELMAP command is:

```
RACDCERT DELMAP[(LABEL('label-name'))]
   [ID(mapping-owner) | MULTIID]
```

If you specify more than one RACDCERT function, only the last specified function is processed. Extraneous keywords that are not related to the function being performed are ignored.

If you do not specify a RACDCERT function, LIST is the default function.

For information on issuing this command as a RACF TSO command, refer to Chapter 3, “RACF TSO commands,” on page 15.

Parameters

**DELMAP**

**DELMAP**(LABEL('label-name'))

Specifying label-name is required if more than one mapping is associated with the user ID. Note that mappings might also be deleted as part of DELUSER processing.

**Important:** If the user profile for the specified user ID no longer exists and you specify a label name, RACF searches all profiles in the DIGTNMAP class to locate and delete the orphaned DIGTNMAP profile. (An orphaned DIGTNMAP profile might result when a DELUSER command is issued from a downlevel system for a user ID that has an associated mapping.) This search might take an extended period of time.

**ID**(mapping-owner) | **MULTIID**

Specifies the user ID associated with the mapping. If you do not specify ID or MULTIID, the default is ID, and mapping-owner defaults to the user ID of the command issuer. If more than one keyword is specified, the last specified keyword is processed and the others are ignored by TSO command parse processing.

**ID**(mapping-owner)

Specifies the user ID associated with the mapping.

**MULTIID**

Specifies that additional criteria is used to determine the user ID associated with the mapping.
Examples

Table 42. RACDCERT DELMAP examples

Example 1

Operation  User RACFADM with SPECIAL authority has been notified that departments BWVB and BWVA have merged. The members of BWVA will be issued new digital certificates.

Known  User RACFADM has SPECIAL authority.

Command  RACDCERT DELMAP (LABEL('BWVA USERS'))
          ID(BWVAUSR)

Output  None.
RACDCERT DELRING

RACDCERT DELRING (Delete key ring)

Purpose

Use the RACDCERT DELRING command to delete a key ring.

Issuing options

The following table identifies the eligible options for issuing the RACDCERT DELRING command:

<table>
<thead>
<tr>
<th>As a RACF TSO command?</th>
<th>As a RACF operator command?</th>
<th>With command direction?</th>
<th>With automatic command direction?</th>
<th>From the RACF parameter library?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>No</td>
<td>No. (See rules.)</td>
<td>No. (See rules.)</td>
<td>No</td>
</tr>
</tbody>
</table>

Rules:

- The RACDCERT command cannot be directed to a remote system using the AT or ONLYAT keyword.
- The updates made to the RACF database by RACDCERT are eligible for propagation with automatic direction of application updates based on the RRSFCDATA profiles AUTODIRECT.target-node.DIGTCERT.APPL and AUTODIRECT.target-node.DIGTRING.APPL, where target-node is the remote node to which the update is to be propagated.

Authorization required

To issue the RACDCERT DELRING command, you must have the SPECIAL attribute or sufficient authority to the IRR.DIGTCERT.DELRING resource in the FACILITY class for your intended purpose.

Table 43. Authority required for the RACDCERT DELRING function

<table>
<thead>
<tr>
<th>IRR.DIGTCERT.DELRING</th>
<th>Access level</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>READ</td>
<td>Delete your own key ring.</td>
<td></td>
</tr>
<tr>
<td>UPDATE</td>
<td>Delete another user’s key ring.</td>
<td></td>
</tr>
</tbody>
</table>

Activating your changes

If the DIGTRING class is RACLISTed, refresh the class to activate your changes.

Example:

```
SETROPTS RACLIST(DIGTRING) REFRESH
```

Related commands

- To add a key ring, see RACDCERT ADDRING.
- To list a key ring, see RACDCERT LISTRING.

Syntax

For the key to the symbols used in the command syntax diagrams, see "Syntax of RACF commands and operands" on page 9. The complete syntax of the RACDCERT DELRING command is:
RACDCERT DELRING

RACDCERT DELRING(ring-name)
  [ID(ring-owner)]

If you specify more than one RACDCERT function, only the last specified function is processed. Extraneous keywords that are not related to the function being performed are ignored.

If you do not specify a RACDCERT function, LIST is the default function.

For information on issuing this command as a RACF TSO command, refer to Chapter 3, “RACF TSO commands,” on page 15.

Parameters

DELRING(ring-name)
The ring-name value is the name of the key ring. Since only user IDs can have key rings, neither CERTAUTH nor SITE can be specified with DELRING.

Note that when a DELUSER command is issued against a user ID, all of the key rings that are owned by that user ID are also deleted.

ID(ring-owner)
Specifies the user ID of the key ring owner. (Only a user ID can have a key ring.) If not specified, the key ring owner defaults to the command issuer’s user ID.

Examples

Table 44. RACDCERT DELRING examples

Example 1

Operation User RACFADM wants to delete the key ring for the local FTP server. The user ID of the FTP is FTPD. The name of the key ring to be deleted is FTPring.

Known User RACFADM has SPECIAL authority.

Command RACDCERT ID(FTPD) DELRING(FTPring)

Output None.
RACDCERT DELTOKEN

RACDCERT DELTOKEN (Delete token)

Purpose

Use the RACDCERT DELTOKEN command to delete a z/OS PKCS #11 token.

Issuing options

The following table identifies the eligible options for issuing the RACDCERT DELTOKEN command:

<table>
<thead>
<tr>
<th>As a RACF TSO command?</th>
<th>As a RACF operator command?</th>
<th>With command direction?</th>
<th>With automatic command direction?</th>
<th>From the RACF parameter library?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>No</td>
<td>No. (See rules.)</td>
<td>No. (See rules.)</td>
<td>No</td>
</tr>
</tbody>
</table>

Rules:

- The RACDCERT command cannot be directed to a remote system using the AT or ONLYAT keyword.
- The updates made to the RACF database by RACDCERT are eligible for propagation with automatic direction of application updates based on the RRSFDATA profiles AUTODIRECT.target-node.DIGTCERT.APPL and AUTODIRECT.target-node.DIGTRING.APPL, where target-node is the remote node to which the update is to be propagated.

Authorization required

Authorization to delete z/OS PKCS #11 tokens is controlled by ICSF based on profiles in the CRYPTOZ class. (No authority in the FACILITY class is required.) If the command issuer does not have authority to delete the specified token as determined by ICSF, the command stops and an error message is displayed.

If the CSFSERV class is active, the command issuer might also need sufficient authority in the CSFSERV class.

For authorization details about the CRYPTOZ and CSFSERV classes, see z/OS Cryptographic Services ICSF Administrator’s Guide.

Related commands

- To add a token, see RACDCERT ADDTOKEN.

Syntax

For the key to the symbols used in the command syntax diagrams, see "Syntax of RACF commands and operands" on page 9. The complete syntax of the RACDCERT DELTOKEN command is:

```
RACDCERT DELTOKEN(token-name)
[FORCE]
```

Note: The ID(certificate-owner) | SITE | CERTAUTH parameter is ignored for this RACDCERT function.

If you specify more than one RACDCERT function, only the last specified function is processed. Extraneous keywords that are not related to the function being performed are ignored.
RACDCERT DELTOKEN

If you do not specify a RACDCERT function, LIST is the default function.

For information on issuing this command as a RACF TSO command, refer to Chapter 3, “RACF TSO commands,” on page 15.

Parameters

DELTOKEN(token-name)
   The token-name value is the name of the token being deleted. If any object within the token is not currently defined to RACF, you must also specify FORCE or else an error message is issued and the command ends. (This error message prevents you from inadvertently deleting a certificate object that is not defined to RACF.)

FORCE
   Specifies that RACF should bypass some error checking and unconditionally perform the delete token operation.

   If you do not specify FORCE, the following condition must be true or an error message is issued and the command ends:
   • The certificate (or its associated private key, if any) must be currently defined to RACF.

   If you specify FORCE, this condition is not checked.

Examples

Table 45. RACDCERT DELTOKEN examples

Example 1
   Operation User ICSFADM has been notified that the z/OS PKCS #11 token named WEBSRV.NETTOKEN is no longer needed and should be deleted.
   Known User ICSFADM has CONTROL authority to the SO.* generic profile in the CRYPTOZ class. The token to be deleted is empty.
   Commands RACDCERT DELTOKEN(WEBSRV.NETTOKEN)
   Output None.
RACDCERT EXPORT

RACDCERT EXPORT (Export certificate package)

Purpose

Use the RACDCERT EXPORT command to write a digital certificate to a data set.

See “UTF-8 and BMP character restrictions” on page 278 for information about how UTF-8 and BMP characters in certificate names and labels are processed by RACDCERT functions.

Issuing options

The following table identifies the eligible options for issuing the RACDCERT EXPORT command:

<table>
<thead>
<tr>
<th>As a RACF TSO command?</th>
<th>As a RACF operator command?</th>
<th>With command direction?</th>
<th>With automatic command direction?</th>
<th>From the RACF parameter library?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>No</td>
<td>No. (See rules.)</td>
<td>No. (See rules.)</td>
<td>No</td>
</tr>
</tbody>
</table>

Rules:
- The RACDCERT command cannot be directed to a remote system using the AT or ONLYAT keyword.
- The updates made to the RACF database by RACDCERT are eligible for propagation with automatic direction of application updates based on the RRSFDATA profiles AUTODIRECT.target-node.DIGTCERT.APPL and AUTODIRECT.target-node.DIGTRING.APPL, where target-node is the remote node to which the update is to be propagated.

Authorization required

To issue the RACDCERT EXPORT command, you must have the SPECIAL attribute or the following authorizations:
- Sufficient authority to the IRR.DIGTCERT.EXPORT resource in the FACILITY class, based on the certificate owner.
- Sufficient authority to the IRR.DIGTCERT.EXPORTKEY resource in the FACILITY class, based on the format of the exported certificate package.

Table 46. Authority required for the RACDCERT EXPORT function

<table>
<thead>
<tr>
<th>Format</th>
<th>Your own certificate</th>
<th>Another user’s certificate</th>
<th>SITE or CERTAUTH certificate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Export in CERT format</td>
<td>Export your own certificate: READ authority to IRR.DIGTCERT.EXPORT</td>
<td>Export another user’s certificate: UPDATE authority to IRR.DIGTCERT.EXPORT</td>
<td>Export SITE or CERTAUTH certificates: CONTROL authority to IRR.DIGTCERT.EXPORT</td>
</tr>
<tr>
<td>Export in PKCS #7 format</td>
<td>Export your own certificate, but not the parent CA chain: READ authority to IRR.DIGTCERT.EXPORT</td>
<td>Export another user’s certificate, but not the parent CA chain: UPDATE authority to IRR.DIGTCERT.EXPORT</td>
<td>Export SITE or CERTAUTH certificates and/or the entire parent CA chain: CONTROL authority to IRR.DIGTCERT.EXPORT</td>
</tr>
<tr>
<td>Export in PKCS #12 format</td>
<td>Export your own certificate and the private key: READ authority to IRR.DIGTCERT.EXPORTKEY</td>
<td>Export another user’s certificate and the private key: CONTROL authority to IRR.DIGTCERT.EXPORTKEY</td>
<td>Export SITE or CERTAUTH certificate and the private key: CONTROL authority to IRR.DIGTCERT.EXPORTKEY</td>
</tr>
</tbody>
</table>
Related commands

- To list a certificate, see RACDCERT LIST.

Syntax

For the key to the symbols used in the command syntax diagrams, see "Syntax of RACF commands and operands" on page 9. The complete syntax of the RACDCERT EXPORT command is:

```
RACDCERT EXPORT(LABEL('label-name'))
   [ID(certificate-owner) | SITE | CERTAUTH]
   DSN(output-data-set-name)
   [FORMAT(
      CERTDER
      | CERTB64
      | PKCS7DER
      | PKCS7B64
      | PKCS12DER
      | PKCS12B64
    )]
   [PASSWORD('pkcs12-password')]
```

If you specify more than one RACDCERT function, only the last specified function is processed. Extraneous keywords that are not related to the function being performed are ignored.

If you do not specify a RACDCERT function, LIST is the default function.

For information on issuing this command as a RACF TSO command, refer to Chapter 3, “RACF TSO commands,” on page 15.

Parameters

**EXPORT(LABEL('label-name'))**

The *label-name* value identifies the certificate that is being exported. Depending on which keyword you specify, you can export a certificate, a certificate and its CA chain, or a certificate and private key.

**ID(certificate-owner) | SITE | CERTAUTH**

Specifies that the specified certificate is either a user certificate associated with the specified user ID, a site certificate, or a certificate-authority certificate. If you do not specify ID, SITE, or CERTAUTH, the default is ID, and *certificate-owner* defaults to the user ID of the command issuer. If more than one keyword is specified, the last specified keyword is processed and the others are ignored by TSO command parse processing.

**DSN(output-data-set-name)**

Specifies the data set that is to contain the certificate. The data set *output-data-set-name* is deleted and reallocated if it exists. If EXPORT is specified, DSN must be specified.

**FORMAT**

Specifies the format of the exported certificate package. Valid values for FORMAT are:

- **CERTB64**—specifies a DER encoded X.509 certificate that has been encoded using Base64.
- **CERTDER**—specifies a DER encoded X.509 certificate.
RACDCERT EXPORT

- **PKCS7B64**—specifies a DER encoded PKCS #7 package that has been encoded using Base64.
- **PKCS7DER**—specifies a DER encoded PKCS #7 package.
- **PKCS12B64**—specifies a DER encoded PKCS #12 package that has been encoded using Base64.
- **PKCS12DER**—specifies a DER encoded PKCS #12 package.

**Notes:**
1. PKCS12DER is the default if PASSWORD is specified; otherwise, CERTB64 is the default.
2. The PKCS12B64 format may not be supported by non-IBM applications. These applications often issue messages indicating incorrect passwords. In this case, reissue the command specifying a format supported by the application.

The CERT keywords indicate that only a certificate is to be exported.

The PKCS #7 keywords indicate to export a certificate and its CA chain. If the command issuer is authorized to export CERTAUTH certificates PKCS #7 processing will attempt to package any certificate authority certificate necessary to complete the basing chain to the exported certificate. If a certificate in the chain cannot be found under CERTAUTH or is expired or the command issuer is not authorize to export CERTAUTH certificates, an informational message will be issued. Processing continues creating an incomplete PKCS #7 package. An incomplete PKCS #7 package can still be processed by RACF but may or may not be useful for OEM products.

The PKCS #12 keywords indicate to export the certificate and the private key (which must exist and must not be an ICSF or PCICC key). The package produced by specifying one of the PKCS #12 keywords is encrypted using the password specified according to the PKCS #12 standard.

PKCS #12 processing requires a software private key. If the private key is stored in ICSF (as an ICSF or PCICC key), the PKCS #12 package cannot be created, and processing stops.

PKCS #12 processing will attempt to package any certificate-authority certificate necessary to complete the basing chain to the exported certificate. If a certificate in the chain cannot be found under CERTAUTH, an informational message will be issued. Processing continues and an incomplete PKCS #12 package is created that can still be processed by RACF but may or may not be useful for OEM products.

**PASSWORD('pkcs12-password')**

Specifies the password to use for PKCS #12 package encryption. The string is converted before being used, so any characters entered must be translatable to 7-bit ASCII. However, RACF does not enforce this.

**Note:** RACF assumes the current host code page is IBM-1047 and translates to ASCII accordingly.
Examples

Table 47. RACDCERT EXPORT examples

Example 1  
**Operation**  User WENTING wants to export a certificate and send it to her business partner Yun.

**Known**  The exported certificate does not contain the private key so the data set Wen Ting transmits to Yun need not be protected in any way.

**Commands**  
```
RACDCERT EXPORT(LABEL('Wen Ting''s certificate'))
  DSN(FOR.YUN.CRT)
```

**Output**  None.
RACDCERT GENCERT

RACDCERT GENCERT (Generate certificate)

Purpose

Use the RACDCERT GENCERT command to create a digital certificate and potentially a public/private key pair.

See “UTF-8 and BMP character restrictions” on page 278 for information about how UTF-8 and BMP characters in certificate names and labels are processed by RACDCERT functions.

Processing details

When you specify an optional request data set containing the PKCS #10 request data, and extensions are present in the request data (not overridden by other keywords specified with the RACDCERT command), they are copied to the certificate being created. These extensions and the logic involved with using them are described in the following tables:

- For subjectKeyIdentifier, see Table 48.
- For authorityKeyIdentifier, see Table 49.
- For keyUsage, see Table 50 on page 329.
- For basicConstraints, see Table 51 on page 329.
- For subjectAltName, see Table 52 on page 329.
- For issuerAltName, see Table 53 on page 329.

Table 48. Logic for the subjectKeyIdentifier extension for GENCERT

<table>
<thead>
<tr>
<th>When the request data set is specified</th>
<th>When the request data set is not specified</th>
</tr>
</thead>
<tbody>
<tr>
<td>The extension is encoded using the subjectKeyIdentifier value from the request data set if present, if not present the extension is encoded by generating the keyIdentifier according to the Public Key Infrastructure Standards.</td>
<td>The extension is encoded by generating the keyIdentifier according to Public Key Infrastructure Standards.</td>
</tr>
</tbody>
</table>

Table 49. Logic for the authorityKeyIdentifier extension for GENCERT

<table>
<thead>
<tr>
<th>When SIGNWITH is specified</th>
<th>When SIGNWITH is not specified</th>
</tr>
</thead>
<tbody>
<tr>
<td>The extension is encoded using the subjectKeyIdentifier value of the signing certificate if present, if not present the extension is not created.</td>
<td>The authorityKeyIdentifier extension is not created.</td>
</tr>
</tbody>
</table>
Table 50. Logic for the keyUsage extension for GENCERT

<table>
<thead>
<tr>
<th>Situation</th>
<th>keyUsage is present in the request data set</th>
<th>keyUsage is not present in the request data set</th>
</tr>
</thead>
<tbody>
<tr>
<td>When KEYUSAGE is specified and the target ID is CERTAUTH</td>
<td>If the certSign bit is turned off in the request data set, the request will fail. Otherwise the extension is encoded as requested by the RACDCERT invoker. Additionally, the certSign and cRLSign bits will be turned on if not already specified by the CERTSIGN keyword.</td>
<td>The extension is encoded as requested by the RACDCERT invoker. Additionally, the certSign and cRLSign bits are turned on.</td>
</tr>
<tr>
<td>When KEYUSAGE is specified and the target ID is SITE or ID(cert-owner)</td>
<td>The extension is encoded as requested by the RACDCERT invoker.</td>
<td>The extension is encoded by turning the certSign and cRLSign bits on.</td>
</tr>
<tr>
<td>When KEYUSAGE is not specified and the target ID is CERTAUTH</td>
<td>If the certSign bit is turned off this command fails, otherwise the extension is encoded as specified in the request data set.</td>
<td>The extension is encoded by turning the certSign and cRLSign bits on.</td>
</tr>
<tr>
<td>When KEYUSAGE is not specified and the target ID is SITE or ID(cert-owner)</td>
<td>The extension is encoded using the request data set values.</td>
<td>The keyUsage extension is not created.</td>
</tr>
</tbody>
</table>

Table 51. Logic for the basicConstraints extension for GENCERT

<table>
<thead>
<tr>
<th>Situation</th>
<th>basicConstraints is present in the request data set</th>
<th>basicConstraints is not present in the request data set</th>
</tr>
</thead>
<tbody>
<tr>
<td>When the target ID is CERTAUTH</td>
<td>If the CA boolean value is false, the command will fail. Otherwise the extension is encoded turning the CA bit on. The pathLength value is not included.</td>
<td>The extension is encoded turning the CA bit on. The pathLength value is not included.</td>
</tr>
<tr>
<td>When the target ID is SITE or ID(cert-owner)</td>
<td>The extension is encoded using the request data set values, including the pathLength value.</td>
<td>The basicConstraints extension is not created.</td>
</tr>
</tbody>
</table>

Table 52. Logic for the subjectAltName extension for GENCERT

<table>
<thead>
<tr>
<th>Situation</th>
<th>subjectAltName is present in the request data set</th>
<th>subjectAltName is not present in the request data set</th>
</tr>
</thead>
<tbody>
<tr>
<td>When ALTNNAME is specified</td>
<td>The extension is encoded as requested by the RACDCERT invoker.</td>
<td>The extension is encoded as requested by the RACDCERT invoker.</td>
</tr>
<tr>
<td>When ALTNNAME is not specified</td>
<td>The extension is encoded using the request data set values.</td>
<td>The subjectAltName extension is not created.</td>
</tr>
</tbody>
</table>

Table 53. Logic for the issuerAltName extension for GENCERT

<table>
<thead>
<tr>
<th>When SIGNWITH is specified</th>
<th>When SIGNWITH is not specified</th>
</tr>
</thead>
<tbody>
<tr>
<td>The extension is encoded using the subjectAltName value of the signing certificate if the extension is present. Otherwise, the issuerAltName extension is not created.</td>
<td>The IssuerAltName extension is not created.</td>
</tr>
</tbody>
</table>

Issuing options

The following table identifies the eligible options for issuing the RACDCERT GENCERT command:
RACDCERT GENCERT

<table>
<thead>
<tr>
<th>As a RACF TSO command?</th>
<th>As a RACF operator command?</th>
<th>With command direction?</th>
<th>With automatic command direction?</th>
<th>From the RACF parameter library?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>No</td>
<td>No. (See rules.)</td>
<td>No. (See rules.)</td>
<td>No</td>
</tr>
</tbody>
</table>

Rules:
- The RACDCERT command cannot be directed to a remote system using the AT or ONLYAT keyword.
- The updates made to the RACF database by RACDCERT are eligible for propagation with automatic direction of application updates based on the RRSFDATA profiles AUTODIRECT.target-node.DIGTCERT.APPL and AUTODIRECT.target-node.DIGTRING.APPL, where target-node is the remote node to which the update is to be propagated.

Authorization required

To issue the RACDCERT GENCERT command, you must have the SPECIAL attribute or the following authorizations:
- Sufficient authority to the IRR.DIGTCERT.ADD resource in the FACILITY class, based on the certificate owner and the SIGNWITH value.
- Sufficient authority to the IRR.DIGTCERT.GENCERT resource in the FACILITY class, based on the certificate owner and the SIGNWITH value.
- If you specify the FROMICSF option, sufficient authority to appropriate resource in the CSFKEYS class. For details about CSFKEYS authorities, see [z/OS Cryptographic Services ICSF Administrator’s Guide](https://www.ibm.com/support/docview.zhtml?docid=2246005).
- READ access to the optional request data set that contains the PKCS #10 request data for the certificate you are adding.

Table 54. Authority required for the RACDCERT GENCERT function

<table>
<thead>
<tr>
<th>SIGNWITH</th>
<th>Your own certificate</th>
<th>Another user’s certificate</th>
<th>SITE or CERTAUTH certificate</th>
</tr>
</thead>
<tbody>
<tr>
<td>SIGNWITH your own certificate</td>
<td>READ authority to IRR.DIGTCERT.ADD and READ authority to IRR.DIGTCERT.GENCERT</td>
<td>UPDATE authority to IRR.DIGTCERT.ADD and READ authority to IRR.DIGTCERT.GENCERT</td>
<td>CONTROL authority to IRR.DIGTCERT.ADD and READ authority to IRR.DIGTCERT.GENCERT</td>
</tr>
<tr>
<td>SIGNWITH a SITE or CERTAUTH certificate</td>
<td>READ authority to IRR.DIGTCERT.ADD and CONTROL authority to IRR.DIGTCERT.GENCERT</td>
<td>UPDATE authority to IRR.DIGTCERT.ADD and CONTROL authority to IRR.DIGTCERT.GENCERT</td>
<td>CONTROL authority to IRR.DIGTCERT.ADD and CONTROL authority to IRR.DIGTCERT.GENCERT</td>
</tr>
<tr>
<td>SIGNWITH not specified</td>
<td>READ authority to IRR.DIGTCERT.ADD and READ authority to IRR.DIGTCERT.GENCERT</td>
<td>UPDATE authority to IRR.DIGTCERT.ADD and UPDATE authority to IRR.DIGTCERT.GENCERT</td>
<td>CONTROL authority to IRR.DIGTCERT.ADD and CONTROL authority to IRR.DIGTCERT.GENCERT</td>
</tr>
</tbody>
</table>

The GENCERT keyword allows a certificate to be generated and signed. Effective controls on the user ID associated with the generated certificate and which certificate is used to sign the generated certificate are essential.

Authority processing details: RACF performs two checks that determine the authority required for the GENCERT command:
1. How the certificate is being signed, specified with the SIGNWITH keyword.
   - Users with SPECIAL authority can use the SIGNWITH keyword with any value.
   - Users without SPECIAL authority must have authority to the IRR.DIGTCERT.GENCERT resource in the FACILITY class. If SIGNWITH is specified without the CERTAUTH or SITE keyword, the certificate is signed with...
the certificate identified with the LABEL keyword for the user who is issuing the RACDCERT command. This requires READ access to the resource IRR.DIGTCERT.GENCERT in the FACILITY class. If either SIGNWITH(CERTAUTH...) or SIGNWITH(SITE) is specified, CONTROL authority is required to the resource IRR.DIGTCERT.GENCERT in the FACILITY class.

Not specifying SIGNWITH indicates that the certificate is to be self-signed. The signing key is owned by the certificate itself. Thus the authority needed for signing is determined by the owner of the generated certificate.

2. What type of certificate is being generated, which is specified with the ID(), SITE or CERTAUTH keywords.

Users with SPECIAL authority can generate a digital certificate for any RACF-defined user or for any certificate-authority or site certificate. Users without SPECIAL authority can generate certificate authority or site certificates if they have CONTROL authority to the resource IRR.DIGTCERT.ADD in the FACILITY class. Users without SPECIAL authority can generate certificates for other users if they have UPDATE authority to the resource IRR.DIGTCERT.ADD in the FACILITY class. Users without SPECIAL authority can generate certificates for themselves if they have READ authority to the resource IRR.DIGTCERT.ADD in the FACILITY class.

Activating your changes

If the DIGTCERT class is RACLISTed, refresh the class to activate your changes.

Example:

SETROPTS RACLIST(DIGTCERT) REFRESH

Related commands

• To add a certificate, see RACDCERT ADD.
• To generate a certificate request, see RACDCERT GENREQ.
• To rekey a expiring certificate, see RACDCERT REKEY.

Syntax

For the key to the symbols used in the command syntax diagrams, see "Syntax of RACF commands and operands" on page 9. The complete syntax of the RACDCERT GENCERT command is:
RACDCERT GENCERT

RACDCERT GENCERT[(request-data-set-name)]

[ID(certificate-owner) | SITE | CERTAUTH]

[SUBJECTSDN(
    [CN('common-name')]
    [T('title')]
    [OU('organizational-unit-name1')
        [ , 'organizational-unit-name2', ... ]]
    [O('organization-name')]
    [L('locality')]
    [SP('state-or-province')]
    [C('country')]
)
)

[SIZE(key-size)]

[NOTBEFORE([DATE(yyyy-mm-dd) [TIME(hh:mm:ss)])

[NOTAFTER([DATE(yyyy-mm-dd) [TIME(hh:mm:ss)])

[WITHLABEL('label-name')]

[SIGNWITH([ CERTAUTH | SITE ] LABEL ('label-name'))]

[(PCICC([pkds-label | *) ]
    [ ICSF([pkds-label | * ) ]
    [ DSA
    [ FROMICSF(pkds-label)]
)

[KEYUSAGE(
    [CERTSIGN]
    [DATAENCRYPT]
    [DOCSIGN]
    [HANDSHAKE]
)

[ALTNAME(
    [IP(numeric-IP-address)]
    [DOMAIN('internet-domain-name')]
    [EMAIL('email-address')]
    [URI('universal-resource-identifier')]
)
]

If you specify more than one RACDCERT function, only the last specified function is processed. Extraneous keywords that are not related to the function being performed are ignored.

If you do not specify a RACDCERT function, LIST is the default function.

For information on issuing this command as a RACF TSO command, refer to Chapter 3, “RACF TSO commands,” on page 15.

Parameters

GENCERT

GENCERT(request-data-set-name)

Request-data-set-name is the name of an optional data set that contains the PKCS #10 certificate request data. The request data contains the user's generated public key and X.509 distinguished name. The request data must be signed, DER-encoded, and then Base64 encoded according to the PKCS #10 standard.

The subkeywords of the GENCERT keyword specify the information that is to be contained within the certificate that is being created.

If request-data-set-name is specified, RACDCERT does not generate a key pair because this data set contains the user's public key.
You cannot specify both request-data-set-name and FROMICSF.

Request-data-set-name has characteristics (for example, RECFM) identical to the data set that can be specified with the ADD and CHECKCERT keywords. If request-data-set-name is specified, SIGNWITH must also be specified because the request-data-set-name data set does not contain a private key. If SIGNWITH is not specified, an informational message is issued. Note that the issuer of the RACDCERT command must have READ access to the request-data-set-name data set to prevent an authorization abend from occurring when the data set is read.

ID(certificate-owner) | SITE | CERTAUTH

Specifies that the new certificate is either a user certificate associated with the specified user ID, a site certificate, or a certificate-authority certificate. If you do not specify ID, SITE, or CERTAUTH, the default is ID, and certificate-owner defaults to the user ID of the command issuer. If more than one keyword is specified, the last specified keyword is processed and the others are ignored by TSO command parse processing.

SUBJECTSDN

Specifies the subject’s X.509 distinguished name, which consists of the following components:

- **CommonName**—specified with the CN subkeyword.
- **Title**—specified with the T subkeyword.
- **Organizational Unit**—specified with the OU subkeyword. Multiple values can be specified for the organizational unit.
- **Organization**—specified with the O subkeyword.
- **Locality**—specified with the L subkeyword.
- **State/Province**—specified with the SP subkeyword.
- **Country**—specified with the C subkeyword.

SUBJECTSDN completely overrides the values contained in the certificate request in the data set specified with the GENCERT keyword.

Each of the elements in SUBJECTSDN is limited to 64 characters. Each of the SUBJECTSDN subkeywords can be specified only once. If the certificate being created is a self-signed certificate, the total length of the subject’s distinguished name must be 229 characters or less. For non-self-signed certificates, the length of the subject’s distinguished name is limited to 255 characters. These lengths include the X.509 identifiers (such as C= and CN=) and the dot qualifiers.

If the SUBJECTSDN name is too long, an informational message is issued and the certificate is not added.

Any printable character that can be mapped to an ASCII character can be specified. Characters that cannot be mapped, such as X'4A' (¢) and X'00' are shown by RACDCERT LIST as blanks.

If SUBJECTSDN and request-data-set-name are not specified, the programmer name data from the ID() user (either specified or defaulted), or the programmer name from the SITE or CERTAUTH anchor user IDs (irrsitec or irrcerta) is used as the common name (CN). If the programmer name is all blanks (X'40'), nulls (X'00'), # characters (X'7B'), or X'FF' characters, the common name is set to the user ID that is to be associated with this certificate.

SIZE(key-size)

Specifies the size of the private key expressed in decimal bits. This keyword is ignored if GENCERT is specified with a request-data-set-name.
Rounding up to the next appropriate key size might occur. Therefore, the key
size of the generated key might be longer than the value you specify with SIZE
but the generated key is never shorter than requested.

If SIZE is not specified, it defaults to 1024. The minimum key size is 512. The
maximum key size is determined by United States export regulations and is
controlled by RACF and non-RACF code in z/OS. Depending on the installation,
non-RACF code may enforce a lower maximum size.

The maximum key size depends on key type:

<table>
<thead>
<tr>
<th>Private key type</th>
<th>Maximum key size</th>
</tr>
</thead>
<tbody>
<tr>
<td>ICSF RSA key</td>
<td>1024 bits</td>
</tr>
<tr>
<td>Non-ICSF DSA key</td>
<td>2048 bits</td>
</tr>
<tr>
<td>Non-ICSF RSA key</td>
<td>4096 bits</td>
</tr>
<tr>
<td>PCI-class cryptographic-coprocessor RSA key</td>
<td>4096 bits</td>
</tr>
</tbody>
</table>

Currently, standard key sizes are as follows:

<table>
<thead>
<tr>
<th>Key size</th>
<th>Key strength</th>
</tr>
</thead>
<tbody>
<tr>
<td>512 bits</td>
<td>Low-strength key</td>
</tr>
<tr>
<td>1024 bits</td>
<td>Medium-strength key</td>
</tr>
<tr>
<td>2048 bits</td>
<td>High-strength key</td>
</tr>
<tr>
<td>4096 bits</td>
<td>Very high-strength key</td>
</tr>
</tbody>
</table>

**NOTBEFORE**(DATE(yyyy-mm-dd) TIME(hh:mm:ss))

Specifies the local date and time from which the certificate is valid. If
DATE(yyyy-mm-dd) is not specified, it defaults to the current local date. If
TIME(hh:mm:ss) is not specified, it defaults to TIME (00:00:00). Note that the
value of yyy must be 1950–2040.

The time and date values are stored in the certificate as a universal time
coordinated (UTC) value. The calculated UTC value can be incorrect if the
NOTBEFORE/NOTAFTER date and time values represent a time that has a
different local offset from UTC.

Note that the use of the date format yyy-mm-dd is valid. However, to aid
installations familiar with the RACF date format, the value can be specified in
the format yyyymm/dd.

**NOTAFTER**(DATE(yyyy-mm-dd) TIME(hh:mm:ss))

Specifies the local date and time after which the certificate is no longer valid. If
DATE(yyyy-mm-dd) is not specified it defaults to one year from the
NOTBEFORE(DATE()) value. If TIME(hh:mm:ss) is not specified, it defaults to
TIME(23:59:59). Note that the value of yyy must be 1950–2040 if specified,
and 1951–2041 if defaulted.

The time and date values are stored in the certificate as a universal time
coordinated (UTC) value. The calculated UTC value can be incorrect if the
NOTBEFORE/NOTAFTER date and time value represent a time that has a
different local offset from UTC.

The NOTBEFORE value must be earlier than the NOTAFTER value or an
informational message is issued.
Note the use of the date format `yyyy-mm-dd` is valid. However, to aid installations familiar with the RACF date format, the value can be specified as `yyyy/mm/dd`.

**WITHLABEL('label-name')**
Specifies the label assigned to this certificate. If specified, this must be unique to the user ID with which the certificate is associated. If not specified, it defaults in the same manner as the WITHLABEL keyword on the RACDCERT ADD command.

The `label-name` is stripped of leading and trailing blanks. If a single quotation mark is intended to be part of the `label-name`, use two single quotation marks together for each single quotation mark within the string, and enclose the entire string within single quotation marks.

See the WITHLABEL keyword for RACDCERT ADD for information on label rules.

**SIGNWITH(CERTAUTH LABEL('label-name'))**
**SIGNWITH(SITE LABEL('label-name'))**
**SIGNWITH(LABEL('label-name'))**
Specifies the certificate with a private key that is signing the certificate. If not specified, the default is to sign the certificate with the private key of the certificate that is being generated. This creates a self-signed certificate. The signing certificate must belong to the user ID executing the command, or SITE or CERTAUTH. If SITE and CERTAUTH keywords are omitted, the signing certificate owner defaults to the user ID of the command issuer.

If SIGNWITH is specified, it must refer to a certificate that has a private key associated with it. If no private key is associated with the certificate, an informational message is issued and processing stops.

If you specify either `request-data-set-name` or FROMICSF, you must specify SIGNWITH.

Note that self-signed certificates are always trusted, while all other certificates are created with the trust status of the certificate specified in the SIGNWITH keyword. If the certificate specified in the SIGNWITH keyword is not trusted, an informational message is issued but the certificate is still generated.

**PCICC | ICSF | DSA | FROMICSF**
Specifies if RACF should generate a new key pair, and if so, how to generate the key pair and where to store the private key for future use. The default action for a new key is to store it as a software key. If no keyword is specified, the key pair is generated using software with the RSA algorithm and the private key is stored in the RACF database as a non-ICSF RSA key.

**PCICC[(pkds-label | *)]**
Specifies that the key pair is to be generated using a PCI-class cryptographic coprocessor. The resulting private key is generated with the RSA algorithm and stored in the ICSF PKA key data set (PKDS) as an ICSF RSA Chinese Remainder Theorem (CRT) key token.

**PCICC system requirements:**
- When specifying PCICC:
  - The ICSF subsystem must be operational and configured for PKA operations.
  - A PCI-class cryptographic coprocessor must also be operational and configured with this PKDS.
**Note:** The Crypto Express2 coprocessor feature is required to process keys longer than 2048 bits. Otherwise, command processing stops and an error message is displayed.

- When the private key is stored as a PCICC key, any system or SSL application using the key must have:
  - An operational ICSF subsystem configured for PKA operations with this PKDS.
  - An operational PCI-class cryptographic coprocessor configured with this PKDS.

**ICSF** *(pkds-label | *)*

Specifies that the key pair is to be generated using software. The resulting private key is generated with the RSA algorithm and stored in the ICSF PKA key data set (PKDS) as an ICSF RSA Modulus-Exponent (ME) key token.

**ICSF system requirements:**

- When specifying ICSF, the ICSF subsystem must be operational and configured for PKA operations. Otherwise, command processing stops and an error message is displayed.
- When the private key is stored as an ICSF key, any system or SSL application using the key must have ICSF operational and configured for PKA operations with this PKDS.

**DSA**

Specifies that the key pair is to be generated using software with DSA algorithm and the private key is stored in the RACF database as a non-ICSF DSA key. **Note:** DSA key generation can be very slow, especially for keys longer than 1024 bits.

**FROMICSF** *(pkds-label)*

Specifies that no new key pair is to be generated for this new certificate. Instead, RACF uses an existing public key specified by its PKDS label. The public key must reside in the ICSF PKA key data set (PKDS).

When you specify FROMICSF, you must also specify SIGNWITH to sign the new certificate with an existing certificate. The new certificate will contain no private key and therefore cannot be self-signed.

You cannot specify both `request-data-set-name` and FROMICSF.

**Guidelines for choosing PCICC, ICSF, or DSA:**

- Choose the ICSF or PCICC keyword when the private key to be generated is RSA, and you want hardware protection for the key.
  - The PCICC key type provides the best performance and supports RSA key sizes up to 4096 bits, but requires a PCI-class cryptographic coprocessor.
  - The ICSF key type can be used on a PCI-class cryptographic coprocessor or the older cryptographic coprocessor facility. However, the key size is limited to 1024 bits.
- Choose DSA when the key to be generated is DSA. Note that no hardware protection is possible for DSA keys.
- Do not specify a key type keyword (neither PCICC, ICSF nor DSA) when the key to be generated is RSA but no hardware protection is desired. Such software keys can be up to 4096 bits in size.
PKDS label considerations for the PCICC and ICSF keywords:

- The PKDS label for the key is optional. You can specify asterisk (*) to use the PKDS label from the WITHLABEL keyword. If you specify an asterisk (*), you must specify the WITHLABEL keyword.

- Whether specified or taken from the WITHLABEL keyword, the PKDS label must be unique and meet ICSF syntax requirements. That is, allowed characters are alphanumeric, national (θ, #, $), or period (.). The first character must be alphabetic or national. The label must be 1–64 characters and is translated to uppercase (not case-sensitive).

If GENCERT is issued without a request-data-set-name value:

- For private keys only, when you do not specify a PKDS label and you do not specify an asterisk (*), RACF generates a default label in the format IRR.DIGTCERT.certificate-owner.cvtsname.ebcdic-stck-value, where certificate-owner is the owning user ID, cvtsname is the system name (taken from the CVT), and ebcidic-stck-value is an EBCDIC version of the current store-clock value. RACF does not generate a PKDS label for a public key.

If GENCERT is issued with a request-data-set-name value:

- No key-pair is generated.

- The public key from the request data set is used in the generated certificate.

- If the FROMICSF keyword is also specified, the GENCERT command fails.

- The DSA keyword, if specified, is ignored.

- The ICSF or PCICC keyword, if specified, is ignored unless one of the following conditions is true:
  - The certificate profile (containing the private key of the corresponding public key) in the request data set already exists and the private key is a non-ICSF key. When this occurs, RACF stores the private key in the ICSF PKDS.
  - There is no corresponding private key profile and you specified a PKDS label value. When this occurs, RACF stores the public key in the ICSF PKDS.

- If the certificate in the request data set already exists, key processing for the ICSF and PCICC keywords continues as indicated.

- If the certificate you are generating has an existing public or private key in the PKDS:
  - The PKDS label of the existing key is not changed when you respecify the label with the ICSF or PCICC keyword.
  - If the private key already exists in the PKDS as an ICSF key, specifying PCICC will not convert the key to a PCICC key.
  - If the private key already exists in the PKDS as a PCICC key, specifying ICSF will not convert the key to an ICSF key.

KEYUSAGE

Specifies the appropriate values for the keyUsage certificate extension, of which one or more of the values might be coded. For certificate authority certificates, the default is CERTSIGN and is always set. There is no default for certificates that are not certificate-authority certificates.

HANDSHAKE

Facilitates identification and key exchange during security handshakes, such as SSL, which set the digitalSignature and keyEncipherment indicators if the key algorithm is RSA. If key type is DSA, keyUsage HANDSHAKE only sets the digitalSignature indicator.
DATAENCRYPT
Encrypts data, which sets the dataEncipherment indicator. If the key type is DSA, this keyUsage is not allowed.

DOCSIGN
Specifies a legally binding signature, which sets the nonRepudiation indicator.

CERTSIGN
Specifies a signature for other digital certificates and CRLs, which sets the keyCertSign and cRLSign indicators.

ALTFNAME
Specifies the appropriate values for the subjectAltName extension, of which one or more of the values might be coded. If required for the extension, RACF converts the entered values to ASCII.

Note: RACF assumes the terminal code page is IBM-1047 and translates to ASCII accordingly.

IP(numeric-IP-address)
Specifies a fully qualified numeric IP address in IPv4 or IPv6 form.
IPv4 dotted decimal form consists of four decimal numbers (each number must be a value from 0–255) separated by periods:
Example: 9.117.2.45
IPv6 form consists of eight 16-bit blocks of hexadecimal characters separated by colons:
In IPv6 form, leading zeros in each block are optional. Successive blocks of zeros can be represented by a single occurrence of ::
Example: 2001:DB8::8:800:200C:417A
An IPv6 address can contain an IPv4 address:
Example: 0:0:0:0:0:ABCD:1.2.3.4

DOMAIN('internet-domain-name')
Specifies a quoted string containing a fully qualified 'internet-domain-name' (such as 'www.widgits.com'). RACF does not check this value's validity.

EMAIL('email-address')
Specifies a quoted string containing a fully qualified 'email-address', such as 'jasper at moes.bar.com'. RACF replaces the word at with the @ symbol (X'7C') to conform with RFC822. If RACF cannot locate the word at it assumes the address is already in RFC822 form and makes no attempt to alter it other than converting it to ASCII.

URI('universal-resource-identifier')
Specifies the 'universal-resource-identifier' (such as 'http://www.widgits.com'). RACF does not check the validity of this value.
Examples

Table 55. RACDCERT GENCERT examples

Example 1
Operation User RACFADM requests the creation of a certificate-authority certificate, with values for the subjectAltName extension and the keyUsage extension, for the local certificate authority. The certificate will be self-signed and a SIGNWITH value need not be specified.

Known User RACFADM has CONTROL access authority to the IRR.DIGTCERT.* resource in the FACILITY class.

Command
RACDCERT GENCERT
CERTAUTH
SUBJECTDN(CN('Local CA'))
ALTNAME(IP(9.117.170.150) DOMAIN('www.widgits.com'))
EMAIL('localca@www.widgits.com')
URI("http://www.widgits.com/welcome.html")
KEYUSAGE(HANDSHAKE DATAENCRYPT DOCSIGN CERTSIGN)
WITHLABEL('Local PKIX CA')

Output None.

Example 2
Operation User WENTING wants to create a new certificate with a 2048-bit public/private key pair so she can share encrypted data with a business partner. She wants to call her certificate Wen Ting's certificate.

Known IBM Encryption Facility requires a PKDS label so RACF creates a default PKDS label.

Command
RACDCERT GENCERT
SUBJECTDN(CN('Wen Ting's certificate'))
WITHLABEL('Wen Ting's certificate')
SIZE(2048)
PCICC
NOTAFTER(DATE(2020/08/10))

Output None. (See the PKDS label generated by RACF using the RACDCERT LIST command in [Figure 38 on page 351].)

Example 3
Operation User RACFADM wants to create a CA certificate that can be used to issue code-signing certificates for users who need to digitally sign programs.

Known User RACFADM has CONTROL access authority to the IRR.DIGTCERT.* resource in the FACILITY class, and appropriate authority in the CSFSERV and CSFKEYS classes to be able to use the PCICC option.

Command
RACDCERT CERTAUTH GENCERT
SUBJECTDN(OU('MyCompany Code Signing CA') O('MyCompany') C('US'))
SIZE(2048) PCICC WITHLABEL('MyCompany Code Signing CA')

Output None.

Example 4
Operation User RACFADM wants to issue a code-signing certificate to user RAMOS who needs to digitally sign programs. The new code-signing certificate will be signed by the CA certificate created in Example 3.

Known User RACFADM has CONTROL access authority to the IRR.DIGTCERT.* resource in the FACILITY class.

Command
RACDCERT ID(RAMOS) GENCERT
SUBJECTDN(CN('Ramos Code Signing Cert') O('MyCompany') C('US'))
SIZE(1024) SIGNWITH(CERTAUTH LABEL('MyCompany Code Signing CA'))
KEYUSAGE(HANDSHAKE DOCSIGN)

Output None.
RACDCERT GENREQ

RACDCERT GENREQ (Generate request)

Purpose

Use the RACDCERT GENREQ command to create a PKCS #10 Base64-encoded certificate request based on the specified certificate and write the request to a data set.

To renew an existing certificate, the GENREQ syntax is RACDCERT GENREQ(LABEL('label-name')) DSN('output-data-set-name'), where label-name is the name of the certificate the request is based on. The request generated does not have a name.

Rule: GENREQ requires that the certificate have a private key associated with it. If no private key is associated with the certificate, an informational message is issued and processing stops.

The certificate request contains the subject’s distinguished name and public key, and is signed with the private key associated with the specified certificate. Additionally, the following extensions are copied to the certificate request if they are present in the certificate:
- subjectAltName
- subjectKeyIdentifier
- authorityKeyIdentifier
- basicConstraints
- keyUsage
- extKeyUsage.

Typically, these requests are sent to a certificate authority; however, they can also be imported into and signed by RACF using the GENCERT keyword with a request-data-set-name.

See “UTF-8 and BMP character restrictions” on page 278 for information about how UTF-8 and BMP characters in certificate names and labels are processed by RACDCERT functions.

Issuing options

The following table identifies the eligible options for issuing the RACDCERT GENREQ command:

<table>
<thead>
<tr>
<th>As a RACF TSO command?</th>
<th>As a RACF operator command?</th>
<th>With command direction?</th>
<th>With automatic command direction?</th>
<th>From the RACF parameter library?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>No</td>
<td>No. (See rules.)</td>
<td>No. (See rules.)</td>
<td>No</td>
</tr>
</tbody>
</table>

Rules:
- The RACDCERT command cannot be directed to a remote system using the AT or ONLYAT keyword.
- The updates made to the RACF database by RACDCERT are eligible for propagation with automatic direction of application updates based on the RRSFDAuto redirect profiles AUTODIRECT.target-node.DIGTCERT.APPL and AUTODIRECT.target-node.DIGTRING.APPL, where target-node is the remote node to which the update is to be propagated.
Authorization required

To issue the RACDCERT GENREQ command, you must have the SPECIAL attribute or sufficient authority to the IRR.DIGTCERT.GENREQ resource in the FACILITY class for your intended purpose.

Table 56. Authority required for the RACDCERT GENREQ function

<table>
<thead>
<tr>
<th>IRR.DIGTCERT.GENREQ</th>
<th>Access level</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Read</td>
<td>READ</td>
<td>Generate a request based on your own certificate.</td>
</tr>
<tr>
<td>Update</td>
<td>UPDATE</td>
<td>Generate a request based on another user’s certificate.</td>
</tr>
<tr>
<td>Control</td>
<td>CONTROL</td>
<td>Generate a request based on a SITE or CERTAUTH certificate.</td>
</tr>
</tbody>
</table>

Related commands

- To add a certificate, see RACDCERT ADD.
- To generate a certificate, see RACDCERT GENCERT.

Syntax

For the key to the symbols used in the command syntax diagrams, see “Syntax of RACF commands and operands” on page 9. The complete syntax of the RACDCERT GENREQ command is:

```
RACDCERT GENREQ(LABEL(‘label-name’))
   [ID(certificate-owner) | SITE | CERTAUTH]
   DSN(output-data-set-name)
```

If you specify more than one RACDCERT function, only the last specified function is processed. Extraneous keywords that are not related to the function being performed are ignored.

If you do not specify a RACDCERT function, LIST is the default function.

For information on issuing this command as a RACF TSO command, refer to Chapter 3, “RACF TSO commands,” on page 15.

Parameters

- **GENREQ(LABEL(‘label-name’))**
  Specifies the label of the certificate used to build the certificate request.

- **ID(certificate-owner) | SITE | CERTAUTH**
  Specifies that the specified certificate is either a user certificate associated with the specified user ID, a site certificate, or a certificate-authority certificate. If you do not specify ID, SITE, or CERTAUTH, the default is ID, and certificate-owner defaults to the user ID of the command issuer. If more than one keyword is specified, the last specified keyword is processed and the others are ignored by TSO command parse processing.

- **DSN(output-data-set-name)**
  Specifies the data set that is to contain the certificate request. The data set output-data-set-name is deleted and reallocated if it exists. If you specify GENREQ, DSN must be specified.
RACDCERT GENREQ

Examples

Table 57. RACDCERT GENREQ examples

Example 1  
**Operation**  User WEBADM needs to create a certificate request based on the expiring certificate for a Web server application, and store it in an MVS data set called 'SYSADM.CERT.REQ'. The user ID of the application is WEBSERV01 and its expiring certificate is labeled 'My Web Server Cert'.

**Known**  User WEBADM has UPDATE access authority to the IRR.DIGTCERT.GENREQ resource in the FACILITY class.

**Command**  
```
RACDCERT GENREQ(LABEL('My Web Server Cert'))
    ID(WEBSRV01)
    DSN('SYSADM.CERT.REQ')
```

**Output**  None.
RACDCERT IMPORT (Import certificate)

Purpose

Use the RACDCERT IMPORT command to import a digital certificate (with its associated private key, if present) from a z/OS PKCS #11 token and add it to RACF.

The IMPORT function processes certificates in the same way as the ADD function with respect to re-adding and renewing certificates, and determining the trust status of certificates.

See “UTF-8 and BMP character restrictions” on page 278 for information about how UTF-8 and BMP characters in certificate names and labels are processed by RACDCERT functions.

Issuing options

The following table identifies the eligible options for issuing the RACDCERT IMPORT command:

<table>
<thead>
<tr>
<th>As a RACF TSO command?</th>
<th>As a RACF operator command?</th>
<th>With command direction?</th>
<th>With automatic command direction?</th>
<th>From the RACF parameter library?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>No</td>
<td>No. (See rules.)</td>
<td>No. (See rules.)</td>
<td>No</td>
</tr>
</tbody>
</table>

Rules:

- The RACDCERT command cannot be directed to a remote system using the AT or ONLYAT keyword.
- The updates made to the RACF database by RACDCERT are eligible for propagation with automatic direction of application updates based on the RRSFDATA profiles AUTODIRECT.target-node.DIGTCERT.APPL and AUTODIRECT.target-node.DIGTRING.APPL, where target-node is the remote node to which the update is to be propagated.

Authorization required

To issue the RACDCERT IMPORT command, you must have the following authorizations:

- The SPECIAL attribute, or sufficient authority to the IRR.DIGTCERT.ADD resource in the FACILITY class based on the certificate owner.
- Sufficient authority to the appropriate resources in the CRYPTOZ class.
- If the CSFSERV class is active, sufficient authority to appropriate resources in the CSFSERV class.

For details about CRYPTOZ and CSFSERV authorities, see z/OS Cryptographic Services ICSF Administrator’s Guide

If the command issuer is not authorized by ICSF (through the CRYPTOZ class) to access the specified token or is not authorized by RACF (through the FACILITY class) to add the specified RACF certificate, the command stops and an error message is displayed.

Table 58. Authority required for the RACDCERT IMPORT function

<table>
<thead>
<tr>
<th>Your own certificate</th>
<th>Another user’s certificate</th>
<th>SITE or CERTAUTH certificate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sufficient authority to CRYPTOZ resources and READ authority to IRR.DIGTCERT.ADD</td>
<td>Sufficient authority to CRYPTOZ resources and UPDATE authority to IRR.DIGTCERT.ADD</td>
<td>Sufficient authority to CRYPTOZ resources and CONTROL authority to IRR.DIGTCERT.ADD</td>
</tr>
</tbody>
</table>
RACDCERT IMPORT

Activating your changes

If the DIGTCERT class is RACLISTed, refresh the class to activate your changes.

Example:
SETROPTS RACLIST(DIGTCERT) REFRESH

Related commands

- To add a certificate, see RACDCERT ADD.

Syntax

For the key to the symbols used in the command syntax diagrams, see "Syntax of RACF commands and operands" on page 9. The complete syntax of the RACDCERT IMPORT command is:

```
RACDCERT IMPORT(TOKEN(token-name) SEQNUM(sequence-number))
   [ ID(certificate-owner) | SITE | CERTAUTH ]
   [ WITHLABEL('label-name') ]
   [ TRUST | NOTRUST | HIGHTRUST ]
   [ PCICC([pkds-label | *]) | ICSF([pkds-label | *]) ]
```

If you specify more than one RACDCERT function, only the last specified function is processed. Extraneous keywords that are not related to the function being performed are ignored.

If you do not specify a RACDCERT function, LIST is the default function.

For information on issuing this command as a RACF TSO command, refer to Chapter 3, “RACF TSO commands,” on page 15.

Parameters

**IMPORT**

**TOKEN(token-name) SEQNUM(sequence-number)**

**TOKEN(token-name)**

Specifies the name of the token from which the certificate is being imported. When specifying the IMPORT operand, you must specify the TOKEN operand.

**SEQNUM(sequence-number)**

Specifies the sequence number of the certificate being imported from the token. When specifying the IMPORT operand, you must specify the SEQNUM operand.

**ID(certificate-owner) | SITE | CERTAUTH**

Specifies that the target owner for the imported certificate is the specified user ID, a site certificate, or a certificate-authority certificate. If you do not specify ID, SITE, or CERTAUTH, the default is ID, and certificate-owner defaults to the user ID of the command issuer. If more than one keyword is specified, the last specified keyword is processed and the others are ignored by TSO command parse processing.
WITHLABEL('label-name')

Specifies the label to be associated with the imported certificate. Up to 32 characters can be specified. The label-name can contain blanks and mixed-case characters.

This label is used as a handle instead of the serial number and issuer's distinguished name. It can be used to store a descriptive text.

If the value specified in WITHLABEL already exists, RACDCERT returns a message indicating that the label has already been used. The certificate is not added.

If WITHLABEL is not specified, RACDCERT generates a label for the certificate. The generated label is of the form LABELnnnnnnnn, where nnnnnnnn is the first integer value, starting at 00000001 that generates a unique label name.

The label-name is stripped of leading and trailing blanks. If a single quotation mark is intended to be part of the label-name, use two single quotation marks together for each single quotation mark within the string, and enclose the entire string within single quotation marks.

PCICC[(pkds-label | *)] | ICSF[(pkds-label | * )]

Specifies that RACF should store the public or private key associated with this certificate in the ICSF PKA key data set (PKDS). This applies when the key is introduced to RACF and when an existing certificate profile is replaced.

The default action for a new key is for RACF to store it as a software key (in the RACF database, not in the ICSF PKDS). The default action for an existing key is to leave it unchanged.

PCICC[(pkds-label | * )]

When you specify PCICC with a PKDS label or an asterisk (*):

- If the certificate has a private key, PCICC specifies that the private key is to be converted using a PCI-class cryptographic coprocessor to an ICSF RSA Chinese Remainder Theorem (CRT) key token. The resulting private key is stored in the ICSF PKDS.
- If the certificate has no private key, PCICC specifies that the public key is to be stored as an ICSF RSA Modulus-Exponent (ME) key token.

If the certificate has no private key and you specify PCICC without a PKDS label and without an asterisk (*), the PCICC keyword is ignored. (No PKDS entry is created.)

PCICC system requirements:

- When specifying PCICC:
  - The ICSF subsystem must be operational and configured for PKA operations.
  - A PCI-class cryptographic coprocessor must also be operational and configured with this PKDS.

  Note: The Crypto Express2 coprocessor feature is required to process keys longer than 2048 bits.

Otherwise, command processing stops and an error message is displayed.

- When the private key is stored as a PCICC key, any system or SSL application using the key must have:
  - An operational ICSF subsystem configured for PKA operations with this PKDS.
AN operational PCI-class cryptographic coprocessor configured with this PKDS.

ICSF([pkds-label | * ])

Specifies that the public or private key is to be converted to an ICSF RSA Modulus-Exponent (ME) key token. The resulting key is stored in the ICSF PKDS.

If the certificate has no private key and you specify ICSF without a PKDS label and without an asterisk (*), the ICSF keyword is ignored. (No PKDS entry is created.)

ICSF system requirements:
- When specifying ICSF, the ICSF subsystem must be operational and configured for PKA operations. Otherwise, command processing stops and an error message is displayed.
- When the private key is stored as an ICSF key, any system or SSL application using the key must have ICSF operational and configured for PKA operations with this PKDS.

Guidelines:
- Choose the ICSF or PCICC keyword when the private key is RSA, and you want hardware protection for the key.
  - The PCICC key type provides the best performance and supports RSA key sizes up to 4096 bits, but requires a PCI-class cryptographic coprocessor.
  - The ICSF key type can be used on a PCI-class cryptographic coprocessor or the older cryptographic coprocessor facility. However, the key size is limited to 1024 bits.
- Choose neither the ICSF nor PCICC keyword when no hardware protection is desired, or when the private key is DSA.

PKDS label considerations for both the PCICC and ICSF keywords:
- The PKDS label for the key is optional. You can specify asterisk (*) to use the PKDS label from the WITHLABEL keyword. If you specify an asterisk (*), you must specify the WITHLABEL keyword.
- Whether specified or taken from the WITHLABEL keyword, the PKDS label must be unique and meet ICSF syntax requirements. That is, allowed characters are alphanumeric, national ($@#), or period (.). The first character must be alphabetic or national. The label must be 1–64 characters and is translated to uppercase (not case-sensitive).
- For private keys only, when you do not specify a PKDS label and you do not specify an asterisk (*), RACF generates a default label in the format IRR.DIGTCERT.certificate-owner.cvtsname.ebcdic-stck-value, where certificate-owner is the owning user ID, cvtsname is the system name (taken from the CVT), and ebcnic-stck-value is an EBCDIC version of the current store-clock value. RACF does not generate a PKDS label for a public key.

If the certificate you are adding has an associated private key: The PCICC or ICSF keyword is ignored and no ICSF PKCS entry is created under any one of these conditions:
- The private key uses the DSA algorithm.
- The private key has already been stored as an ICSF or PCICC key.

If the certificate you are adding has an existing public or private key in the PKDS, and the key is already associated with this certificate:
RACDCERT IMPORT

- If the public key already exists in the PKDS and you add its matching private key certificate, you must specify the ICSF or PCICC keyword. (This upgrades the PKDS entry to include the new ICSF or PCICC private key.)
- The PKDS label of the existing public or private key is not changed when you re-specify the label with the ICSF or PCICC keyword.
- If the private key already exists in the PKDS as an ICSF key, specifying PCICC will not convert the key to a PCICC key.
- If the private key already exists in the PKDS as a PCICC key, specifying ICSF will not convert the key to an ICSF key.

If the certificate you are adding has an existing public or private key in the PKDS but the key is not already associated with this (or another) RACF certificate: RACF associates the key with this certificate when all of the following conditions are true:
- You specify the PKDS label of the existing key using the ICSF or PCICC keyword.
- The existing key is either an RSA encrypted private key (an ICSF internal key token) or an RSA public key.
- No private key is already associated with this certificate.

Examples

Table 59. RACDCERT IMPORT examples

Example 1

Operation
User NETB0Y wishes to add a digital certificate to RACF and associate it with his own user ID. The certificate is labeled Savings Account and currently resides in the z/OS PKCS #11 token named NETB0Y.TKN1. The status of the certificate will be trusted.

Known
User NETB0Y has READ authority to the discrete profile named IRR.DIGTCERT.ADD in the FACILITY class, and READ authority to the discrete profile named USER.NETB0Y.TKN1 in the CRYPTOZ class. Using RACDCERT LISTTOKEN, user NETB0Y determined the sequence number of the certificate to be added is 3.

Command
RACDCERT IMPORT(TOKEN(NETB0Y.TKN1) SEQNUM(3))
   ID(NETB0Y) TRUST WITHLABEL('Savings Account')

Output
None.
RACDCERT LIST

RACDCERT LIST (List certificate)

Purpose

Use the RACDCERT LIST command to display digital certificate information, including certificate authority and site certificate information. You can also use the RACDCERT LIST command to list all certificates owned by a user ID.

Because the virtual key ring for a user ID consists of all certificates owned by the user ID, using the RACDCERT LIST command to list all certificates owned by a user ID is the same as listing the contents of the virtual key ring for that user ID.

For each digital certificate defined, the following information is displayed:

- Label
- Certificate ID
- Status (trusted, not trusted, or highly trusted)
- Validity dates
- Serial number
- Issuer's distinguished name
- Up to 256 bytes of the subject's name, as found in the certificate itself
- Extensions, if present (specifically, keyUsage and subjectAltName)
- Type of private key (ICSF, non-ICSF, PCICC, or none)
- PKDS label (if the public or private key is stored in the ICSF PKA key data set (PKDS)
- Private key size
- Ring associations, if present (the ring name to which this certificate is connected and the ring owner)

See "UTF-8 and BMP character restrictions" on page 278 for information about how UTF-8 and BMP characters in certificate names are displayed using RACDCERT functions.

Issuing options

The following table identifies the eligible options for issuing the RACDCERT LIST command:

<table>
<thead>
<tr>
<th>As a RACF TSO command?</th>
<th>As a RACF operator command?</th>
<th>With command direction?</th>
<th>With automatic command direction?</th>
<th>From the RACF parameter library?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>No</td>
<td>No (See rules.)</td>
<td>No (See rules.)</td>
<td>No</td>
</tr>
</tbody>
</table>

Rules:

- The RACDCERT command cannot be directed to a remote system using the AT or ONLYAT keyword.
- The updates made to the RACF database by RACDCERT are eligible for propagation with automatic direction of application updates based on the RRSFDATA profiles AUTODIRECT.target-node.DIGTCERT.APPL and AUTODIRECT.target-node.DIGTRING.APPL, where target-node is the remote node to which the update is to be propagated.
Authorization required

To issue the RACDCERT LIST command, you must have the SPECIAL attribute or sufficient authority to the IRR.DIGTCERT.LIST resource in the FACILITY class for your intended purpose.

Table 60. Authority required for the RACDCERT LIST function

<table>
<thead>
<tr>
<th>Access level</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>READ</td>
<td>List your own certificate.</td>
</tr>
<tr>
<td>UPDATE</td>
<td>List another user's certificate.</td>
</tr>
<tr>
<td>CONTROL</td>
<td>List SITE or CERTAUTH certificates.</td>
</tr>
</tbody>
</table>

Related commands

- To list a key ring, see RACDCERT LISTRING.
- To list a token, see RACDCERT LISTTOKEN.

Syntax

For the key to the symbols used in the command syntax diagrams, see "Syntax of RACF commands and operands" on page 9. The complete syntax of the RACDCERT LIST command is:

```
RACDCERT [ LIST
    [ (LABEL('label-name')) ]
    [ (SERIALNUMBER(serial-number) [ ISSUERSDN('issuer’s-dn') ] ) ]
    [ ID(certificate-owner) | SITE | CERTAUTH ]
```

If you specify more than one RACDCERT function, only the last specified function is processed. Extraneous keywords that are not related to the function being performed are ignored.

If you do not specify a RACDCERT function, LIST is the default function.

For information on issuing this command as a RACF TSO command, refer to Chapter 3, "RACF TSO commands," on page 15.

Parameters

- `LIST(LABEL('label-name'))`
- `LIST(SERIALNUMBER(serial-number) ISSUERSDN('issuer’s-dn'))`

If the RACDCERT command is issued with no other operands, LIST is the default and the RACDCERT command lists the command issuer's digital certificate information. If the RACDCERT command is issued with the ID keyword and no other operands, it lists the digital certificate information associated with the user ID specified with the ID keyword.

The issuer’s distinguished name and the subject’s distinguished name can contain blanks. If the name displayed in the output is subsequently entered with the ISSUERSDN keyword, the blanks must be included. In the output of LIST, the characters > and < are used to mark the beginning and end of the serial
RACDCERT LIST

number, issuer’s name, and subject’s name. When information continues to the next line, < appears in column 79 of the output, and > appears in column 9 of the continuation line.

If the user has only one certificate, or if all certificates are to be displayed, the SERIALNUMBER and ISSUERSDN keywords, or the LABEL keyword, and their associated values can be omitted. If the user has more than one certificate the LABEL, SERIALNUMBER, or SERIALNUMBER and ISSUERSDN can be used to select which certificate to list.

**Note:** The ISSUERSDN and LABEL keywords are case and blank sensitive, therefore, the values specified for these keywords need to be exact when being specified.

For a description of label-name, see the description of the WITHLABEL keyword for RACDCERT ADD.

If present, the SubjectAltName values are displayed under the heading **Subject’s AltNames.** The subheadings **IP, EMail, Domain,** and **URI** are followed by the appropriate values. If more than one line is required to display the value, the additional lines will start in the same column. The word at replaces the @ symbol for email-address. For example,


If present, the keyUsage values are displayed next to the heading **Key Usage.** The possible values are:

- **HANDSHAKE,** which indicates digitalSignature and keyEncipherment are on
- **DATAENCRIPT,** which indicates dataEncipherment is on
- **DOCSIGN,** which indicates nonRepudiation is on
- **CERTSIGN,** which indicates keyCertSign and cRLSign is on

The keyUsage values are displayed as GENCERT keywords separated by commas. For example:

**Key Usage:** HANDSHAKE, CERTSIGN

**Note:** Original RACF digital certificate support did not allow for labels to be specified. These certificates display **No label assigned** in the label field when listed.

**ID(certificate-owner) | SITE | CERTAUTH**

Specifies that the certificate to list is either a user certificate associated with the specified user ID, a site certificate, or a certificate-authority certificate. If you do not specify ID, SITE, or CERTAUTH, the default is ID, and **certificate-owner** defaults to the user ID of the command issuer. If more than one keyword is specified, the last specified keyword is processed and the others are ignored by TSO command parse processing.
Examples

Table 61. RACDCERT LIST examples

Example 1

Operation  User NETB0Y requests the listing of his Savings Account digital certificate to ensure it has been defined, and that it is marked trusted. He has READ authority to the FACILITY class profile IRR.DIGTCERT.LIST. He issues the RACDCERT command with the LIST keyword, specifying the label to identify his certificate.

Known  User NETB0Y has been given READ access to profile IRR.DIGTCERT.LIST in the FACILITY class.

Command  RACDCERT LIST(LABEL('Savings Account'))

Output  See Figure 40 on page 352

Example 2

Operation  User GEORGEM requests the listing of all certificates associated with his user ID.

Known  User ID GEORGEM has 3 certificates, one of which is not associated with any rings.

Command  RACDCERT LIST

Output  See Figure 41 on page 353

Example 3

Operation  User CADUDE wants to list the information from the local certificate-authority certificate with HIGHTRUST status.

Known  User CADUDE has CONTROL authority to the profile IRR.DIGTCERT.* in the FACILITY class.

Command  RACDCERT ID(CERTAUTH LIST(LABEL('Local PKIX CA'))

Output  See Figure 42 on page 354

Example 4

Operation  User CADUDE wants to list information from the certificate of user MSURESH.

Known  User CADUDE has CONTROL authority to the profile IRR.DIGTCERT.* in the FACILITY class. User SURESH has only one certificate. The certificate is self-signed and was issued by the Show Me The € Bank. Because the Euro symbol (€) does not map to the IBM-1047 code page, the certificate listing contains the Euro symbol represented by six characters in the format U+20AC, where 20AC is the hexadecimal form of the Unicode code point for the Euro symbol.

Command  RACDCERT ID(MSURESH) LIST

Output  See Figure 43 on page 354

RACDCERT LIST(LABEL('Wen Ting''s certificate'))

Digital certificate information for user WENTING:
Label: Wen Ting's certificate
Certificate ID: 2Q1hXkbZx8XUaqweQMOFmNA46ixhIBgQOKFmUB7QPDw
Status: TRUST
Start Date: 2005/08/11 00:00:00
End Date: 2020/08/10 23:59:59
Serial Number: >00<
Issuer's Name:
>CN=Wen Ting's certificate<
Subject's Name:
>CN=Wen Ting's certificate<
Private Key Type: PCICC
Private Key Size: 2048
PKDS Label: IRR.DIGTCERT.WENTING.SY1.BD7103108611F42F

Figure 38. Sample output from the RACDCERT LIST command showing an assigned PKDS label
RACDCERT SITE LIST(LABEL('WenTing'))

Digital certificate information for SITE:
  Label: WenTing
  Certificate ID: egljcv8XUaqweQMOFmaNA46iXhUBgQOKFmUB7QP0w
  Status: TRUST
  Start Date: 2005/08/11 00:00:00
  End Date: 2020/08/10 23:59:59
  Serial Number:
  >00<
  Issuer's Name:
  >CN=Wen Ting's certificate<
  Subject's Name:
  >CN=Wen Ting's certificate<
  Private Key Type: None
  PKDS Label: WENTING

Figure 39. Sample output from the RACDCERT LIST command showing a specified PKDS label

RACDCERT LIST(LABEL('Savings Account'))

Digital certificate information for user NETB0Y:
  Label: Savings Account
  Certificate ID: 2QbVxePCIjigaWjiYeioMGDg5aklA
  Status: TRUST
  Serial Number:
  >5D666C20207A6638727A413B8413B<
  Issuer's Name:
  >OU=BobsBank Savers.O=BobsBank.L=Internet<
  Subject's Name:
  >CN=S.S.Smith.OU=Digital ID Class 1 - Netscape.OU=BobsBank Class 1 - S<
  avingsAcct.O=BobsBank.L=Internet<

Figure 40. Sample output from the RACDCERT LIST command specifying the certificate by label
#### RACDCERT LIST

Digital certificate information for user GEORGEM:

<table>
<thead>
<tr>
<th>Certificate ID</th>
<th>Status</th>
<th>Start Date</th>
<th>End Date</th>
<th>Serial Number</th>
<th>Issuer's Name</th>
<th>Subject's Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>2QfHxdbzb2x8XU1YWmQ0OFl4iXhUBgQOKFmUB7QP0w</td>
<td>TRUST</td>
<td>1996/04/18 03:01:13</td>
<td>1998/02/13 03:01:13</td>
<td>&gt;00&lt;</td>
<td>OU=Internet Demo CertAuth.O=The Cert Software Inc.&lt;</td>
<td>OU=Internet Demo CertAuth.O=The Cert Software Inc.&lt;</td>
</tr>
<tr>
<td>2QfHxdbzb2x8XU1YWmQ0OFl4iXhUBgQ0WiiYeVw/FA</td>
<td>TRUST</td>
<td>1998/04/22 23:23:26</td>
<td>2001/01/15 23:23:26</td>
<td>&gt;3511A55290FE7D315E32B31AC8548F0&lt;</td>
<td>OU=Class 1 Public Primary Certification Authority.O=VeriSign, Inc..C=US</td>
<td>OU=VeriSign Class 1 CertAuth - Individual Subscriber.O=VeriSign, Inc..L=Internet</td>
</tr>
<tr>
<td>2QfHxdbzb2x8XU1YWmQ0OFl4iXhUBgQ0WiiYeVw/JA</td>
<td>NOTRUST</td>
<td>1999/03/19 15:39:52</td>
<td>1999/03/19 15:39:52</td>
<td>&gt;50D35294912F79D315E32B31AC8548F0&lt;</td>
<td>OU=Class 2 Public Primary Certification Authority.O=VeriSign, Inc..C=US</td>
<td>OU=VeriSign Class 2 CertAuth - Individual Subscriber.O=VeriSign, Inc..L=Internet</td>
</tr>
</tbody>
</table>

Private Key Type: ICSF
Private Key Type: Non-ICSF
Private Key Size: 1024
Private Key Size: 512

**Ring Associations:**
- Ring Owner: GEORGEM
  - Ring: GEORGEMSNewRing01
  - Ring: GEORGEMSNewRing01
- Ring Owner: GEORGEM
  - Ring: GEORGEMSNewRing01
  - Ring: GEORGEMSNewRing01
- Ring Owner: GEORGEM
  - Ring: GEORGEMSNewRing01
  - Ring: GEORGEMSNewRing01

Label: New Cert Type - Ser # 00
Label: New Type Cert - VsignC1
Label: New Type Cert - VsignC2

PKDS Label: IRR.DIGTCERT.GEORGEM.SY1.8D7103106611842F

Figure 41. Sample output from the RACDCERT LIST command listing all certificates owned by the command issuer.
Digital certificate information for CERTAUTH:

Label: Local PKIX CA
Certificate ID: Sc9bjZwLxKw2myumP1Ny8lGzJQSYi/u35j0eyFe213XgGBMTsUvCW
Status: HIGHTRUST
Start Date: 1999/08/05 00:00:00
End Date: 2000/08/05 23:59:59
Serial Number:
>00<
Issuer's Name:
>CN=Local CA<
Subject's Name:
>CN=Local CA<
Subject's AltNames:
IP: 9.117.170.150
EMail: localca at www.widgits.com
Domain: www.widgits.com
URI: http://www.widgits.com/welcome.html
Key Usage: HANDSHAKE, DATAENCRIPT, DOCSIGN, CERTSIGN
Private Key Type: Non-ICSF
Private Key Size: 1024
Ring Associations:
*** No rings associated ***

Figure 42. Sample output from the RACDCERT LIST command for a CERTAUTH certificate

Digital certificate information for user MSURESH:

Label: Euro
Certificate ID: 2QfJwTk4sXZxaSZ1kB
Status: NOTRUST
Start Date: 2008/10/04 00:00:00
End Date: 2020/01/01 00:00:00
Serial Number:
>68655BB4D15C5F8D45ED01BC51EBD7<
Issuer's Name:
>CN=Show Me The U+20AC Bank<
Subject's Name:
>CN=Show Me The U+20AC Bank<
Private Key Type: None
Ring Associations:
*** No rings associated ***

Figure 43. Sample output from the RACDCERT LIST command showing a UTF-8 or BMP character that does not map to the IBM-1047 code page
RACDCERT LISTMAP (List mapping)

Purpose

Use the RACDCERT LISTMAP command to display information about the specified mapping, or all mappings, for a user ID.

See “UTF-8 and BMP character restrictions” on page 278 for information about how UTF-8 and BMP characters in certificate names are displayed using RACDCERT functions.

Issuing options

The following table identifies the eligible options for issuing the RACDCERT LISTMAP command:

<table>
<thead>
<tr>
<th>As a RACF TSO command?</th>
<th>As a RACF operator command?</th>
<th>With command direction?</th>
<th>With automatic command direction?</th>
<th>From the RACF parameter library?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>No</td>
<td>No. (See rules.)</td>
<td>No. (See rules.)</td>
<td>No</td>
</tr>
</tbody>
</table>

Rules:

- The RACDCERT command cannot be directed to a remote system using the AT or ONLYAT keyword.
- The updates made to the RACF database by RACDCERT are eligible for propagation with automatic direction of application updates based on the RRSFCDATA profiles AUTODIRECT.target-node.DIGTCRIT.APPL and AUTODIRECT.target-node.DIGTCRIT.APPL, where target-node is the remote node to which the update is to be propagated.

Authorization required

To issue the RACDCERT LISTMAP command, you must have the SPECIAL attribute or sufficient authority to the IRR.DIGTCERT.LISTMAP resource in the FACILITY class for your intended purpose.

Table 62. Authority required for the RACDCERT LISTMAP function

<table>
<thead>
<tr>
<th>IRR.DIGTCERT.LISTMAP</th>
<th>Access level</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>READ</td>
<td>List mapping information associated with your own user ID.</td>
<td></td>
</tr>
<tr>
<td>UPDATE</td>
<td>List mapping information associated with another user ID or MULTIID.</td>
<td></td>
</tr>
</tbody>
</table>

Related commands

- To define a user ID mapping, see RACDCERT MAP.
- To alter a user ID mapping, see RACDCERT ALTMAP.
- To delete a user ID mapping, see RACDCERT DELMAP.

The RACDCERT LISTMAP command is unrelated to the RACMAP LISTMAP command.

Syntax

For the key to the symbols used in the command syntax diagrams, see “Syntax of RACF commands and operands” on page 9. The complete syntax of the
RACDCERT LISTMAP command is:

```
RACDCERT LISTMAP((LABEL('label-name')))
   [ ID(mapping-owner) | MULTIID ]
```

If you specify more than one RACDCERT function, only the last specified function is processed. Extraneous keywords that are not related to the function being performed are ignored.

If you do not specify a RACDCERT function, LIST is the default function.

For information on issuing this command as a RACF TSO command, refer to Chapter 3, “RACF TSO commands,” on page 15.

Parameters

**LISTMAP**

**LISTMAP(LABEL('label-name'))**

*Tip:* Omit LABEL to list all mappings associated with the specified user ID.

If the mapping cannot be listed because the DIGTNMAP profile containing it is missing or incomplete, the following error text appears in the LISTMAP output:

```
Filter with label label-name not found.
```

*Guideline:* When this error text appears in the LISTMAP output, issue a RACDCERT DELMAP command specifying this label name to remove residual filter information from the user’s profile.

A missing or incomplete DIGTNMAP profile might result if a previous RACDCERT command failed to complete due to a system failure or early termination by the issuer. If the mapping or DIGTNMAP profile were not created before the failure, the resulting user profile might contain residual filter information indicating that the user ID is associated with a mapping.

**ID(mapping-owner) | MULTIID**

Specifies the user ID associated with the mapping. If you do not specify ID or MULTIID, the default is ID, and mapping-owner defaults to the user ID of the command issuer. If more than one keyword is specified, the last specified keyword is processed and the others are ignored by TSO command parse processing.

**ID(mapping-owner)**

Specifies the user ID associated with the mapping.

**MULTIID**

Specifies that additional criteria is used to determine the user ID associated with the mapping.

Examples

*Table 63. RACDCERT LISTMAP examples*

**Example 1**

**Operation** User RACFADM with SPECIAL authority to the profile

IRR.DIGTCERT.LISTMAP would like to list the mapping information for user ID NET1ID.

**Known** NET1ID has one mapping associated with it.

**Command** RACDCERT ID(NET1ID) LISTMAP

**Output** See Figure 44 on page 357
Table 63. RACDCERT LISTMAP examples (continued)

Example 2

Operation  User RACFADM with SPECIAL authority to the profile IRR.DIGTCERT.LISTMAP would like to list the mapping information for MULTIID.

Known  MULTIID has several mappings associated with it, but only the one with this label name will be listed.

Command  RACDCERT MULTIID LISTMAP(LABEL('NewAPPL ID Mapping'))

Output  See Figure 45

Mapping information for user NET1ID:

Label: General Internet ID Map
Status: TRUST
Issuer's Name Filter:
  >OU=Internet Demo CertAuth.O=Bob's Mart Software Inc.L=Internet<
Subject's Name Filter:
  >L=Internet<

Figure 44. Sample output from the LISTMAP command

Mapping information for MULTIID:

Label: NewAPPL ID Mapping
Status: TRUST
Issuer's Name Filter:
  >OU=Class 1 Public Primary Certification Authority.O=VeriSign, Inc..C=<
Subject's Name Filter:
  ><
Criteria:
  APPLID=APPLID

Figure 45. Sample output from the LISTMAP LABEL command
RACDCERT LISTRING

RACDCERT LISTRING (List key ring)

Purpose

Use the RACDCERT LISTRING command to display the specified key ring, or all key rings, associated with a user, certificate authority, or site certificate.

Issuing options

The following table identifies the eligible options for issuing the RACDCERT LISTRING command:

<table>
<thead>
<tr>
<th>As a RACF TSO command?</th>
<th>As a RACF operator command?</th>
<th>With command direction?</th>
<th>With automatic command direction?</th>
<th>From the RACF parameter library?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>No</td>
<td>No. (See rules.)</td>
<td>No. (See rules.)</td>
<td>No</td>
</tr>
</tbody>
</table>

Rules:

- The RACDCERT command cannot be directed to a remote system using the AT or ONLYAT keyword.
- The updates made to the RACF database by RACDCERT are eligible for propagation with automatic direction of application updates based on the RRSFDATA profiles AUTODIRECT.target-node.DIGTCERT.APPL and AUTODIRECT.target-node.DIGTRING.APPL, where target-node is the remote node to which the update is to be propagated.

Authorization required

To issue the RACDCERT LISTRING command, you must have the SPECIAL attribute or sufficient authority to the IRR.DIGTCERT.LISTRING resource in the FACILITY class for your intended purpose.

Table 64. Authority required for the RACDCERT LISTRING function

<table>
<thead>
<tr>
<th>IRR.DIGTCERT.LISTRING</th>
<th>Access level</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>READ</td>
<td>List your own key ring.</td>
<td></td>
</tr>
<tr>
<td>UPDATE</td>
<td>List another user’s key ring.</td>
<td></td>
</tr>
</tbody>
</table>

Related commands

- To list a certificate, see RACDCERT LIST.
- To list a token, see RACDCERT LISTTOKEN.

Syntax

For the key to the symbols used in the command syntax diagrams, see "Syntax of RACF commands and operands" on page 9. The complete syntax of the RACDCERT LISTRING is:

```
RACDCERT LISTRING[(ring-name)]
   [ID(ring-owner)]
```

If you specify more than one RACDCERT function, only the last specified function is processed. Extraneous keywords that are not related to the function being performed are ignored.
If you do not specify a RACDCERT function, LIST is the default function.

For information on issuing this command as a RACF TSO command, refer to Chapter 3, “RACF TSO commands,” on page 15.

Parameters

LISTRING

LISTRING(ring-name | *)

The ring-name value is the name of the key ring. To list all rings that are associated with a particular user, LISTRING(*) must be specified. For each certificate in the ring, the following information is displayed:

- The ring name
- The owner of the certificate (ID(name), CERTAUTH, or SITE)
- The label assigned to the certificate
- The DEFAULT status of the certificate within the ring
- The usage within the ring.

Since only user IDs can have key rings, neither CERTAUTH nor SITE can be specified with LISTRING.

ID(ring-owner)

Specifies the user ID of the key ring owner. (Only a user ID can have a key ring.) If not specified, the key ring owner defaults to the command issuer's user ID.

Examples

Table 65. RACDCERT LISTRING examples

Example 1

Operation User GEORGEM requests the listing of his key rings.

Known User ID GEORGEM has three key rings with certificates and one key ring that has no certificates.

Command RACDCERT LISTRING(*)

Output See Figure 46 on page 360
Digital ring information for user GEORGEM:

<table>
<thead>
<tr>
<th>Certificate Label Name</th>
<th>Cert Owner</th>
<th>USAGE</th>
<th>DEFAULT</th>
</tr>
</thead>
<tbody>
<tr>
<td>New Cert Type - Ser # 00</td>
<td>ID(GEORGEM)</td>
<td>PERSONAL</td>
<td>YES</td>
</tr>
<tr>
<td>New Type Cert - VsignCl</td>
<td>ID(GEORGEM)</td>
<td>CERTAUTH</td>
<td>NO</td>
</tr>
<tr>
<td>New Type Cert - VsignC2</td>
<td>ID(GEORGEM)</td>
<td>SITE</td>
<td>NO</td>
</tr>
<tr>
<td>65</td>
<td>ID(JOHNP)</td>
<td>PERSONAL</td>
<td>NO</td>
</tr>
</tbody>
</table>

Ring:

<table>
<thead>
<tr>
<th>Certificate Label Name</th>
<th>Cert Owner</th>
<th>USAGE</th>
<th>DEFAULT</th>
</tr>
</thead>
<tbody>
<tr>
<td>GEORGEM's Cert # 48</td>
<td>ID(GEORGEM)</td>
<td>PERSONAL</td>
<td>NO</td>
</tr>
<tr>
<td>GEORGEM's Cert # 84</td>
<td>ID(GEORGEM)</td>
<td>PERSONAL</td>
<td>NO</td>
</tr>
<tr>
<td>New Cert Type - Ser # 00</td>
<td>ID(GEORGEM)</td>
<td>PERSONAL</td>
<td>YES</td>
</tr>
</tbody>
</table>

Ring:

**No certificates connected**

Figure 46. Sample output from the RACDCERT LISTRING command
RACDCERT LISTTOKEN (List token)

Purpose

Use the RACDCERT LISTTOKEN command to display information about the certificate objects in a z/OS PKCS #11 token.

Issuing options

The following table identifies the eligible options for issuing the RACDCERT LISTTOKEN command:

<table>
<thead>
<tr>
<th>As a RACF TSO command?</th>
<th>As a RACF operator command?</th>
<th>With command direction?</th>
<th>With automatic command direction?</th>
<th>From the RACF parameter library?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>No</td>
<td>No. (See rules.)</td>
<td>No. (See rules.)</td>
<td>No</td>
</tr>
</tbody>
</table>

Rules:
- The RACDCERT command cannot be directed to a remote system using the AT or ONLYAT keyword.
- The updates made to the RACF database by RACDCERT are eligible for propagation with automatic direction of application updates based on the RRSFDATA profiles AUTODIRECT.target-node.DIGTCERT.APPL and AUTODIRECT.target-node.DIGTRING.APPL, where target-node is the remote node to which the update is to be propagated.

Authorization required

To issue the RACDCERT LISTTOKEN command, you must have the following authorizations:
- The SPECIAL attribute, or sufficient authority to the IRR.DIGTCERT.LIST resource in the FACILITY class based on the certificate owner.
- Sufficient authority to the appropriate resources in the CRYPTOZ class.
- If the CSFSERV class is active, sufficient authority to appropriate resources in the CSFSERV class.

For details about CRYPTOZ and CSFSERV authorities, see z/OS Cryptographic Services ICSF Administrator's Guide.

If the command issuer is not authorized by ICSF (through the CRYPTOZ class) to read the specified token, the command stops and an error message is displayed. If the command user is authorized to read the specified token but not authorized by RACF (through the FACILITY class) to list the RACF certificates, the output listing contains token information but no certificate information.

Table 66. Authority required for the RACDCERT IMPORT function

<table>
<thead>
<tr>
<th>Your own certificate</th>
<th>Another user’s certificate</th>
<th>SITE or CERTAUTH certificate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sufficient authority to CRYPTOZ resources, and READ authority to IRR.DIGTCERT.LIST</td>
<td>Sufficient authority to CRYPTOZ resources, and UPDATE authority to IRR.DIGTCERT.LIST</td>
<td>Sufficient authority to CRYPTOZ resources, and CONTROL authority to IRR.DIGTCERT.LIST</td>
</tr>
</tbody>
</table>

Related commands

- To list a certificate, see RACDCERT LIST.
- To list a key ring, see RACDCERT LISTRING.
RACDCERT LISTTOKEN

Syntax

For the key to the symbols used in the command syntax diagrams, see "Syntax of RACF commands and operands" on page 9. The complete syntax of the RACDCERT LISTTOKEN is:

```
RACDCERT LISTTOKEN(token-name | *)
```

Note: The ID(certificate-owner) | SITE | CERTAUTH parameter is ignored for this RACDCERT function.

If you specify more than one RACDCERT function, only the last specified function is processed. Extraneous keywords that are not related to the function being performed are ignored.

If you do not specify a RACDCERT function, LIST is the default function.

For information on issuing this command as a RACF TSO command, refer to Chapter 3, “RACF TSO commands,” on page 15.

Parameters

```
LISTTOKEN(token-name | *)
```

To list all tokens that the command issuer is authorized to access, specify LISTTOKEN (*).

For each certificate object in the token that the command issuer is authorized to access with at least READ authority, the following information is displayed:

- The token name
- The sequence number of the certificate object in the token
- The DEFAULT status of the certificate within the token
- The status indicating whether the certificate has an associated private key
- The status indicating whether the certificate has an associated public key
- The certificate’s usage within the token (PERSONAL, SITE or CERTAUTH)
- The ICSF token data set (TKDS) label assigned to the certificate object.
- If the certificate is installed in RACF, the RACF label of the certificate.
- If the certificate is installed in RACF, the owner of the certificate is listed as one of the following values:
  - ID(certificate-owner)
  - CERTAUTH
  - SITE

Examples

Table 67. RACDCERT LISTTOKEN examples

Example 1

Operation The security administrator wants to display information for all certificate objects in the z/OS PKCS #11 token called VENDOR.TOKEN.

Known Commands RACDCERT LISTTOKEN(VENDOR.TOKEN)

Output See Figure 47 on page 363
RACDCERT LISTTOKEN

<table>
<thead>
<tr>
<th>Seq Num</th>
<th>Attributes</th>
<th>Labels</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Default: YES</td>
<td>TKDS: HTTP Serv</td>
</tr>
<tr>
<td></td>
<td>Priv Key: YES</td>
<td>RACF: Webserver Cert</td>
</tr>
<tr>
<td></td>
<td>Usage: PERSONAL</td>
<td>Owner: ID(WEBSRV)</td>
</tr>
<tr>
<td></td>
<td>Pub Key: YES</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Default: NO</td>
<td>TKDS: Extranet CA</td>
</tr>
<tr>
<td></td>
<td>Priv Key: No</td>
<td>RACF: Extranet CA</td>
</tr>
<tr>
<td></td>
<td>Usage: CERTAUTH</td>
<td>Owner: CERTAUTH</td>
</tr>
<tr>
<td></td>
<td>Pub Key: No</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Default: NO</td>
<td>TKDS: Code signing certificate</td>
</tr>
<tr>
<td></td>
<td>Priv Key: YES</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Usage: PERSONAL</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Pub Key: No</td>
<td></td>
</tr>
</tbody>
</table>

Figure 47. Sample output of RACF details from the RACDCERT LISTTOKEN command
RACDCERT MAP

RACDCERT MAP (Create mapping)

Purpose

Use the RACDCERT MAP command to define a user ID mapping, also called a certificate name filter. Defining a mapping results in the creation of a profile in the DIGTNMAP class. DIGTNMAP profiles are used as filters when a user attempts to access the system using a digital certificate. A user ID is found by comparing the issuer’s distinguished name and subject’s distinguished name from the certificate with the filter values used to create the DIGTNMAP profile. The user ID is specified with the ID keyword or specified in DIGTCRIT profiles if MULTIID is specified. When you specify MAP, you must specify IDNFILTER, SDNFILTER, or both.

See “UTF-8 and BMP character restrictions” on page 278 for information about how UTF-8 and BMP characters in certificate names and labels are processed by RACDCERT functions.

Issuing options

The following table identifies the eligible options for issuing the RACDCERT MAP command:

<table>
<thead>
<tr>
<th>As a RACF TSO command?</th>
<th>As a RACF operator command?</th>
<th>With command direction?</th>
<th>With automatic command direction?</th>
<th>From the RACF parameter library?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>No</td>
<td>No. (See rules.)</td>
<td>No. (See rules.)</td>
<td>No</td>
</tr>
</tbody>
</table>

Rules:

- The RACDCERT command cannot be directed to a remote system using the AT or ONLYAT keyword.
- The updates made to the RACF database by RACDCERT are eligible for propagation with automatic direction of application updates based on the RRSFDATA profiles AUTODIRECT.target-node.DIGTNMAP.APPL and AUTODIRECT.target-node.DIGTCRIT.APPL, where target-node is the remote node to which the update is to be propagated.

Authorization required

To issue the RACDCERT MAP command, you must have the SPECIAL attribute or sufficient authority to the IRR.DIGTCERT.MAP resource in the FACILITY class for your intended purpose.

Table 68. Authority required for the RACDCERT MAP function

<table>
<thead>
<tr>
<th>IRR.DIGTCERT.MAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Access level</td>
</tr>
<tr>
<td>READ</td>
</tr>
<tr>
<td>UPDATE</td>
</tr>
</tbody>
</table>

Activating your changes

If the DIGTNMAP or DIGTCRIT class is RACLISTed, refresh the classes to activate your changes.

Example:

```
SETROPTS RACLIST(DIGTNMAP, DIGTCRIT) REFRESH
```
Related commands

- To alter a user ID mapping, see RACDCERT ALTMAP.
- To delete a user ID mapping, see RACDCERT DELMAP.
- To list a user ID mapping, see RACDCERT LISTMAP.

The RACDCERT MAP command is *unrelated* to the RACMAP MAP command.

Syntax

For the key to the symbols used in the command syntax diagrams, see the Syntax of RACF commands and operands on page 9. The complete syntax of the RACDCERT MAP command is:

```
RACDCERT MAP[[(data-set-name)]]
  [ ID(mapping-owner) | MULTIID ]
  [ SDNFILTER('subject's-distinguished-name-filter') ]
  [ IDNFILTER('issuer's-distinguished-name-filter') ]
  [ CRITERIA(criteria-profile-name-template) ]
  [ WITHLABEL('label-name') ]
  [ TRUST | NOTRUST ]
```

If you specify more than one RACDCERT function, only the last specified function is processed. Extraneous keywords that are not related to the function being performed are ignored.

If you do not specify a RACDCERT function, LIST is the default function.

For information on issuing this command as a RACF TSO command, refer to Chapter 3, “RACF TSO commands,” on page 15.

Parameters

**MAP**

**MAP**(data-set-name)

A data set name can be specified with the MAP keyword. The *data-set-name* value is the name of the data set that contains a certificate. The certificate provides a model for the filter names specified with SDNFILTER and IDNFILTER. The subject’s distinguished name is used beginning with the value specified by SDNFILTER. The issuer’s distinguished name is used beginning with the value specified by IDNFILTER. Using a model certificate is optional but can reduce the chance of typographical errors when entering long filters for SDNFILTER or IDNFILTER.

The model certificate used with the MAP keyword can have an issuer’s distinguished name or subject’s distinguished name that exceeds 255 characters. However, the portion of each used in the filter to associate a user ID with the certificate cannot exceed 255 characters.

See RACDCERT ADD for acceptable certificate formats.

The *data-set-name* value has the same characteristics (for example, RECFM) as the data set that can be specified with the ADD and CHECKCERT keywords. The issuer of the RACDCERT command must have READ access to the data set containing the *data-set-name*. 
ID(mapping-owner) | MULTIID
Specifies the user ID to be associated with the new mapping. If you do not specify ID or MULTIID, the default is ID, and mapping-owner defaults to the user ID of the command issuer. If more than one keyword is specified, the last specified keyword is processed and the others are ignored by TSO command parse processing.

ID(mapping-owner)
Specifies the user ID to be associated with the mapping.

MULTIID
Specifies that additional criteria is used to determine the user ID to be associated with the mapping. You must also specify the CRITERIA keyword.

IDNFILTER('issuer's-distinguished-name-filter')
Specifies the significant portion of the issuer’s distinguished name that is used as a filter when associating a user ID with a certificate. For an explanation of how filter values are used to associate a user ID with a digital certificate, see z/OS Security Server RACF Security Administrator’s Guide.

When specified without data-set-name on the MAP keyword, you must specify the entire portion of the distinguished name to be used as a filter.

The format of the issuer's-distinguished-name-filter is similar to the output displayed when a certificate is listed with RACDCERT. It is an X.509 distinguished name in an address type format:

component.component.component.component ...

Or, more specifically:

qualifier1=node1.qualifier2=node2.qualifier3=node3 ...

Example:


Restriction: The filter name cannot contain the ¢ character (X'4A').

The value specified for IDNFILTER must begin with a prefix found in the following list, followed by an equal sign (X'7E'). Each component should be separated by a period (X'4B'). The case, blanks, and punctuation displayed when the digital certificate information is listed must be maintained in the IDNFILTER. Since digital certificates only contain characters available in the ASCII character set, the same characters should be used for the IDNFILTER value. Valid prefixes are:

<table>
<thead>
<tr>
<th>Prefix</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>Country</td>
<td>Specified as C=</td>
</tr>
<tr>
<td>State/Province</td>
<td>Specified as SP=</td>
</tr>
<tr>
<td>Locality</td>
<td>Specified as L=</td>
</tr>
<tr>
<td>Organization</td>
<td>Specified as O=</td>
</tr>
<tr>
<td>Organizational Unit</td>
<td>Specified as OU=</td>
</tr>
<tr>
<td>Title</td>
<td>Specified as T=</td>
</tr>
<tr>
<td>Common Name</td>
<td>Specified as CN=</td>
</tr>
</tbody>
</table>

When specified along with data-set-name on the MAP keyword, the issuer's-distinguished-name-filter must correspond to a starting point within the issuer’s distinguished name found in the certificate contained in the data set. You should specify enough of the name to precisely identify the starting point for the filter. For example, if the certificate in the data set has the following issuer:

and you want all certificates issued by
BobCA
to be selected by this filter, you specify:
IDNFILTER('O=BobCA')

Without the data set containing the certificate, you need to enter the following to produce the same result:
IDNFILTER('O=BobCA, Inc.L=internet.C=US')

A maximum of 255 characters can be entered for IDNFILTER. When a starting point value is specified for a certificate contained in a data set, there cannot be more than 255 characters between the starting point and the end of the issuer's name in the certificate.

IDNFILTER is optional if SDNFILTER is specified. If IDNFILTER is not specified, only the subject's name is used as a filter. If IDNFILTER is specified and only a portion of the issuer's name is to be used as the filter, SDNFILTER must not be specified.

If both IDNFILTER and SDNFILTER are specified, the IDNFILTER value does not need to begin with a valid prefix from the list above. This allows the use of certificates from a certificate authority that chooses to include nonstandard data in the issuer's distinguished name.

SDNFILTER('subject's-distinguished-name-filter')
Specifies the significant portion of the subject's distinguished name. This is the part of the name that will be used as a filter when associating a user ID with a certificate. For an explanation of how filter values are used to associate a user ID with a digital certificate, see z/OS Security Server RACF Security Administrator's Guide.

When specified without data-set-name on the MAP keyword, you must specify the entire portion of the distinguished name to be used as the filter.

The format of the subject's-distinguished-name-filter is similar to the output displayed when a certificate is listed with RACDCERT. It is an X.509 distinguished name in an address type format:

component.component.component.component ...

Or, more specifically:

qualifier1=node1.qualifier2=node2.qualifier3=node3 ...

For example:

SDNFILTER('CN=Bob Cook.OU=BobsAccounting.O=BobsMart.L=internet')

Restriction: The filter name cannot contain the € character (X'4A').

The value specified for SDNFILTER must begin with a prefix found in the following list, followed by an equal sign (X'7E'). Each component should be separated by a period (X'4B'). The case, blanks, and punctuation displayed when the digital certificate information is listed must be maintained in the SDNFILTER. Since digital certificates only contain characters available in the ASCII character set, the same characters should be used for the SDNFILTER value. Valid prefixes are:

- Country Specified as C=
- State/Province Specified as SP=
- Locality Specified as L=
When specified along with data-set-name on the MAP keyword, the subject's distinguished-name-filter must correspond to a starting point within the subject's distinguished name found in the certificate contained in the data set. You should specify enough of the name to precisely identify the starting point for the filter. For example, if the certificate in the data set has the following subject:

\[\text{CN=Bob Cook.OU=BobsAccounting.O=BobsMart.L=internet}\]

and you want all certificates for anyone in BobsAccounting to be selected by this filter, you specify:

\[\text{SDNFILTER('OU=BobsAcc')}\]

Without the data set containing the certificate, you need to enter the following to produce the same result:

\[\text{SDNFILTER('OU=BobsAccounting.O=BobsMart.L=internet')}\]

A maximum of 255 characters can be entered for SDNFILTER. When a starting point value is specified for a certificate contained in a data set, there cannot be more than 255 characters between the starting point and ending point of the subject's name in the certificate.

SDNFILTER is optional if IDNFILTER is specified. If SDNFILTER is not specified, only the issuer's name is used as a filter. SDNFILTER must not be specified with IDNFILTER unless the value of IDNFILTER will result in the entire issuer's name being used in the filter. Note that subject's name can be partial but cannot be used in a filter that contains only a partial issuer's name.

**CRITERIA(criteria-profile-name-template)**

When specified with MULTIID, it indicates a dynamic user ID mapping. The user ID associated with this mapping profile is based not only on the issuer's distinguished name and the subject's distinguished name found in the certificate, but also on additional criteria. The criteria-profile-name-template specifies the additional criteria in the form of a profile name containing one or more variable names, separated by free-form text. These variable names begin with an ampersand (&) and end with a period. The free-form text should identify the variables contained in the template:

\[\text{variable-name1=}&\text{variable-name1.variable-name2=}&\text{variable-name2}...\]

For example, if the application identity and system identifier are to be considered in determining the user ID associated with this mapping, the CRITERIA keyword should be specified as follows:

\[\text{CRITERIA(APPLID=}&\text{APPLID.SYSID=}&\text{SYSID)}\]

The RACF-defined criteria are the application ID (APPLID) and the system-identifier (SYSID). When a user presents a certificate to the system for identification, the identity of the application (as well as the system the user is trying to access) being accessed becomes part of the criteria. The application passes its identity to RACF, and RACF determines the system-identifier. The
system-identifier is the 4-character value specified for the SID parameter of the SMFPRMxx member of SYS1.PARMLIB. These values are substituted for &APPLID and &SYSID in the criteria.

Once the substitution is made, the fully expanded criteria template is used as a resource name to find a matching profile defined in the DIGTCRIT class using the RDEFINE command. For example, if the application being accessed is BANKU on system SYSA, the template is:

\texttt{APPLID=BANKU.SYSID=SYSA}

You should define a profile in the DIGTCRIT class using the RDEFINE command for this name. The user ID to be associated with these certificates must be specified as the APPLDATA. While the DIGTCRIT profile name can be discrete, generic profiles can be used if you have generic profile checking active for the DIGTCRIT class. A DIGTCRIT profile name of \texttt{APPLID=BANKU.*} allows the certificates to be used on any system, rather than just system SYSA. While generic characters such as * and % can be used when defining the DIGTCRIT class profiles, they should not be used in the template name specified with the CRITERIA keyword.

Criteria names other than APPLID and SYSID are allowed, but are effective in certificate name filtering if the application supplies these criteria names and their associated values to RACF when the user attempts to access the application using a certificate. SYSID is determined by RACF, but APPLID must be specified with the initACEE callable service. Criteria names, such as APPLID and SYSID, should only be specified on RACDCERT if the application instructs you to do so.

A maximum of 255 characters can be entered when specifying the CRITERIA keyword. The values can be entered in any case, but are made uppercase by the RACDCERT command because they must match uppercase profile names in the DIGTCRIT class to be effective. When specifying the criteria value, the maximum length for profile names in the DIGTCRIT class is 246 characters.

The CRITERIA keyword can only be set for MULTIID.

\textbf{WITHLABEL(\textit{label-name})}

Specifies the label that is assigned to this mapping. If specified, it must be unique to the user ID with which the mapping is associated. If WITHLABEL is not specified, a label is generated in the same manner as issuing the WITHLABEL keyword for the RACDCERT ADD command.

Up to 32 characters can be specified for \textit{label-name}. It can contain imbedded blanks and mixed-case characters, and is stripped of leading and trailing blanks. If a single quotation mark is intended to be part of the \textit{label-name}, use two single quotation marks together for each single quotation mark within the string, and enclose the entire string within single quotation marks.

\textbf{TRUST | NOTRUST}

When specified with MAP, indicates whether this mapping can be used to associate a user ID to a certificate presented by a user accessing the system. If neither TRUST nor NOTRUST is specified, the default is TRUST.
Examples

Table 69. RACDCERT MAP examples

Example 1

Operation
User RACFADM with SPECIAL authority requests the addition of a new
mapping profile that will associate the user ID WEBUSER with all digital
certificates issued by VeriSign for Class 1 Individual Subscribers. A certificate
is not readily available in a data set.

Known
User RACFADM has SPECIAL authority.

Command
RACDCERT ID(WEBUSER) MAP
IDNFILTER('OU=VeriSign Class 1 Individual Subscriber.O=VeriSign, Inc..L=Internet')
WITHLABEL('Savings Account')

Output
None.

Example 2

Operation
User RACFADM with SPECIAL authority requests the addition of a new
mapping profile that will associate all members of department BWVA, who
have VeriSign Class 1 Individual subscriber certificates, with the user ID
BWVAUSR. All members of the department have the organizational unit BWVA
(OU=BWVA) as the second node of the subject name in their certificates. A
certificate belonging to one of the department member is available in the data
set JJONES.DEPTCERT. The use of the certificates should not be allowed until
the network administrator gives his approval, so this mapping is currently not
trusted.

Known
User RACFADM has SPECIAL authority to profile IRR.DIGTCERT.MAP in the
FACILITY class.

Command
RACDCERT ID(BWVAUSR) MAP('JJONES.DEPTCERT')
IDNFILTER('OU=VeriSign Class 1') NOTRUST
SDNFILTER('OU=BWVA') WITHLABEL('BWVA USERS')

Output
None.

Example 3

Operation
User CERTADM with ALTER authority to profile IRR.DIGTCERT.MAP in the
FACILITY class has received a digital certificate and placed it in the data set
CERTADM.MODEL.CERT. BobsBank has contracted VeriSign, Inc. to create
certificates like the one received. These certificates will be installed on the
workstations of each bank teller, and used to access the banking application
BANKAPP. All certificates must map to the user ID BANKU which has access
to the data sets containing the banking data. CERTADM uses this function to
display the issuer's name and subject name from the certificate.

Known
User CERTADM has ALTER authority to profiles IRR.DIGTCERT.MAP in the
FACILITY class.

Commands
RDEF DIGTCRIT BOBS.APPLID1=BANKAPP APPLDATA('BANKU')
RACDCERT MULTIID MAP(MODEL.CERT)
IDNFILTER('OU=')
SDNFILTER('CN=')
CRITERIA(BOBS.APPLID1=&APPLID)
WITHLABEL('Bobs Tellers')

Output
None.
RACDCERT REKEY (Rekey certificate)

Purpose

Use the RACDCERT REKEY command to replicate (rekey) a digital certificate with a new public/private key pair. In general, after you rekey a certificate, issue the RACDCERT ROLLOVER command to supersede the old certificate with the new rekeyed certificate and retire the old private key. For sample procedures, see “Renewing a certificate with a new private key (rekeying)” in z/OS Security Server RACF Security Administrator’s Guide.

During rekeying, the subject’s distinguished name, key usage, and subject alternate name are copied from the original certificate to the replicated certificate. The replicated certificate is then self-signed and stored as a new certificate associated with the same user ID, CERTAUTH, or SITE. The original certificate is not changed by this operation.

If the rekeyed certificate needs to be signed by another certificate in RACF or another certificate authority, issue the RACDCERT GENREQ command to create a PKCS #10 request from the replicated certificate. Use the resulting request as input to RACDCERT GENCERT to sign the replicated certificate with another certificate in RACF or sent to the external certificate authority for fulfillment. Perform signing (if needed) before issuing the RACDCERT ROLLOVER command.

See [UTF-8 and BMP character restrictions on page 278] for information about how UTF-8 and BMP characters in certificate names and labels are processed by RACDCERT functions.

Issuing options

The following table identifies the eligible options for issuing the RACDCERT REKEY command:

<table>
<thead>
<tr>
<th>As a RACF TSO command?</th>
<th>As a RACF operator command?</th>
<th>With command direction?</th>
<th>With automatic command direction?</th>
<th>From the RACF parameter library?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>No</td>
<td>No. (See rules.)</td>
<td>No. (See rules.)</td>
<td>No</td>
</tr>
</tbody>
</table>

Rules:
- The RACDCERT command cannot be directed to a remote system using the AT or ONLYAT keyword.
- The updates made to the RACF database by RACDCERT are eligible for propagation with automatic direction of application updates based on the RRSFDATA profiles AUTODIRECT.target-node.DIGTCERT.APPL and AUTODIRECT.target-node.DIGTRING.APPL, where target-node is the remote node to which the update is to be propagated.

Authorization required

To issue the RACDCERT REKEY command, you must have the SPECIAL attribute or sufficient authority to the IRR.DIGTCERT.REKEY resource in the FACILITY class for your intended purpose.
RACDCERT REKEY

Table 70. Authority required for the RACDCERT REKEY function

<table>
<thead>
<tr>
<th>Access level</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>READ</td>
<td>Rekey your own certificate.</td>
</tr>
<tr>
<td>UPDATE</td>
<td>Rekey another user's certificate.</td>
</tr>
<tr>
<td>CONTROL</td>
<td>Rekey a SITE or CERTAUTH certificate.</td>
</tr>
</tbody>
</table>

Activating your changes

If the DIGTCERT class is RACLISTed, refresh the class to activate your changes.

Example:

SETROPTS RACLIST(DIGTCERT) REFRESH

Related commands

- To rollover an expiring certificate to a rekeyed certificate, see RACDCERT ROLLOVER.
- To generate a certificate, see RACDCERT GENCERT.

Syntax

For the key to the symbols used in the command syntax diagrams, see "Syntax of RACF commands and operands" on page 9. The complete syntax of the RACDCERT REKEY command is:

```
RACDCERT REKEY(LABEL('existing-label-name'))
  [ID(certificate-owner) | SITE | CERTAUTH]
  [SIZE(key-size)]
  [NOTBEFORE([DATE(yyyy-mm-dd]) [TIME(hh:mm:ss)]]]
  [NOTAFTER([DATE(yyyy-mm-dd]) [TIME(hh:mm:ss)]]]
  [PCICC([pkds-label | * ] | ICSF([pkds-label | * ] ])
  [WITHLABEL('to-be-created-label-name')]
```

If you specify more than one RACDCERT function, only the last specified function is processed. Extraneous keywords that are not related to the function being performed are ignored.

If you do not specify a RACDCERT function, LIST is the default function.

For information on issuing this command as a RACF TSO command, refer to Chapter 3, “RACF TSO commands,” on page 15.

Parameters

REKEY(LABEL('existing-label-name'))

Specifies the label of the existing certificate to be replicated during RACDCERT REKEY processing. This keyword is required and identifies an existing certificate owned by ID(certificate-owner), SITE, or CERTAUTH. The certificate identified by the LABEL keyword must be associated with the must have a private key associated with it otherwise an error message is issued and the command ends.
The subkeywords of the REKEY keyword specify additional information on how to generate and store the new certificate.

**ID(certificate-owner) | SITE | CERTAUTH**

Specifies that the certificate to replicate is either a user certificate associated with the specified user ID, a site certificate, or a certificate-authority certificate. If you do not specify ID, SITE, or CERTAUTH, the default is ID, and certificate-owner defaults to the user ID of the command issuer. If more than one keyword is specified, the last specified keyword is processed and the others are ignored by TSO command parse processing.

**SIZE(key-size)**

Specifies the size of the new private key expressed in decimal bits.

Rounding up to the next appropriate key size might occur. Therefore, the key size of the generated key might be longer than the value you specify with SIZE but the generated key is never shorter than requested.

If SIZE is not specified, it defaults to 1024. The minimum key size is 512. The maximum key size is determined by United States export regulations and is controlled by RACF and non-RACF code in z/OS. Depending on the installation, non-RACF code may enforce a lower maximum size.

The maximum key size depends on key type:

<table>
<thead>
<tr>
<th>Private key type</th>
<th>Maximum key size</th>
</tr>
</thead>
<tbody>
<tr>
<td>ICSF RSA key</td>
<td>1024 bits</td>
</tr>
<tr>
<td>Non-ICSF DSA key</td>
<td>2048 bits</td>
</tr>
<tr>
<td>Non-ICSF RSA key</td>
<td>4096 bits</td>
</tr>
<tr>
<td>PCI-class crypto-coprocessor RSA key</td>
<td>4096 bits</td>
</tr>
</tbody>
</table>

Currently, standard key sizes are as follows:

<table>
<thead>
<tr>
<th>Key size</th>
<th>Key strength</th>
</tr>
</thead>
<tbody>
<tr>
<td>512 bits</td>
<td>Low-strength key</td>
</tr>
<tr>
<td>1024 bits</td>
<td>Medium-strength key</td>
</tr>
<tr>
<td>2048 bits</td>
<td>High-strength key</td>
</tr>
<tr>
<td>4096 bits</td>
<td>Very high-strength key</td>
</tr>
</tbody>
</table>

**NOTBEFORE(DATE/yyyy-mm-dd) TIME(hh:mm:ss)**

Specifies the local date and time from which the new certificate is valid. If DATE/yyyy-mm-dd is not specified, it defaults to the current local date. If TIME(hh:mm:ss) is not specified, it defaults to TIME (00:00:00). Note that the value of yyyy must be 1950–2040.

The time and date values are stored in the certificate as a universal time coordinated (UTC) value. The calculated UTC value can be incorrect if the NOTBEFORE/NOTAFTER date and time values represent a time that has a different local offset from UTC.

Note that the use of the date format yyyy-mm-dd is valid. However, to aid installations familiar with the RACF date format, the value can be specified in the format yyyy/mm/dd.

**NOTAFTER(DATE/yyyy-mm-dd) TIME(hh:mm:ss)**

Specifies the local date and time after which the new certificate is no longer valid. If DATE/yyyy-mm-dd is not specified it defaults to one year from the
NOTBEFORE(DATE()) value. If TIME(hh:mm:ss) is not specified, it defaults to TIME(23:59:59). Note that the value of yyyy must be 1950–2040 if specified, and 1951–2041 if defaulted.

The time and date values are stored in the certificate as a universal time coordinated (UTC) value. The calculated UTC value can be incorrect if the NOTBEFORE/NOTAFTER date and time value represent a time that has a different local offset from UTC.

The NOTBEFORE value must be earlier than the NOTAFTER value or an informational message is issued.

Note the use of the date format yyyy-mm-dd is valid. However, to aid installations familiar with the RACF date format, the value can be specified as yyyy/mm/dd.

PCICC[pkds-label I * ] | ICSF[pkds-label I * ]
Specifies how RACF should generate the private key and how it should be stored for future use if the key algorithm of the original certificate is RSA. The default action for a new key is to store it as a software key.

PCICC[pkds-label I * ]
Specifies that the private key is to be generated using a PCI-class cryptographic coprocessor. The resulting private key is generated with the RSA algorithm and stored in the ICSF PKA key data set (PKDS) as an ICSF RSA Chinese Remainder Theorem (CRT) key token.

PCICC system requirements:
• When specifying PCICC:
  – The ICSF subsystem must be operational and configured for PKA operations.
  – A PCI-class cryptographic coprocessor must also be operational and configured with this PKDS.

Note: The Crypto Express2 coprocessor feature is required to process keys longer than 2048 bits.

Otherwise, command processing stops and an error message is displayed.
• When the private key is stored as a PCICC key, any system or SSL application using the key must have:
  – An operational ICSF subsystem configured for PKA operations with this PKDS.
  – An operational PCI-class cryptographic coprocessor configured with this PKDS.

ICSF[pkds-label I * ]
Specifies that the private key is to be generated using software. The resulting private key is generated with the RSA algorithm and stored in the ICSF PKA key data set (PKDS) as an ICSF RSA Modulus-Exponent (ME) key token.

ICSF system requirements:
• When specifying ICSF, the ICSF subsystem must be operational and configured for PKA operations. Otherwise, command processing stops and an error message is displayed.
• When the private key is stored as an ICSF key, any system or SSL application using the key must have ICSF operational and configured for PKA operations with this PKDS.
Guidelines:

- Choose the ICSF or PCICC keyword when the private key is RSA, and you want hardware protection for the key.
  - The PCICC key type provides the best performance and supports RSA key sizes up to 4096 bits, but requires a PCI-class cryptographic coprocessor.
  - The ICSF key type can be used on a PCI-class cryptographic coprocessor or the older cryptographic coprocessor facility. However, the key size is limited to 1024 bits.
- Choose neither the ICSF nor PCICC keyword when no hardware protection is desired, or when the private key is DSA.

PKDS label considerations for both the PCICC and ICSF keywords:

- The PKDS label for the key is optional. You can specify asterisk (*) to use the PKDS label from the WITHLABEL keyword. If you specify an asterisk (*), you must specify the WITHLABEL keyword.
- Whether specified or taken from the WITHLABEL keyword, the PKDS label must be unique and meet ICSF syntax requirements. That is, allowed characters are alphanumeric, national (θ, №, $), or period (.). The first character must be alphabetic or national. The label must be 1–64 characters and is translated to uppercase (not case-sensitive).
- For private keys only, when you do not specify a PKDS label and you do not specify an asterisk (*), RACF generates a default label in the format IRR.DIGTCERT:certificate-owner.cvtsname.ebcdic-stck-value, where certificate-owner is the owning user ID, cvtsname is the system name (taken from the CVT), and ebcdic-stck-value is an EBCDIC version of the current store-clock value. RACF does not generate a PKDS label for a public key.

If either keyword is specified and ICSF is not operational or is not configured for PKA operations, processing stops and an error message is displayed.

If neither PCICC nor ICSF keyword is specified:

- If neither keyword is specified and the key algorithm of the original certificate is DSA, the private key is generated using software with the DSA algorithm and the private key is stored in the RACF database as a non-ICSF DSA key.
- If neither keyword is specified and the key algorithm of the original certificate is RSA, the private key is generated using software with the RSA algorithm and the private key is stored in the RACF database as a non-ICSF RSA key.

WITHLABEL('to-be-created-label-name')

Specifies the label assigned to the new certificate. If specified, this must be unique to the user ID with which the certificate is associated. If not specified, it defaults in the same manner as the WITHLABEL keyword on the RACDCERT ADD command.

The label-name value is stripped of leading and trailing blanks. If a single quotation mark is intended to be part of the label-name, use two single quotation marks together for each single quotation mark within the string, and enclose the entire string within single quotation marks.

See the WITHLABEL keyword for RACDCERT ADD for information on label rules.

Note: Use the GENCERT keyword instead of the REKEY keyword if you want to change the subject’s distinguished name or supported extensions in addition to creating a new key pair.
RACDCERT REKEY

Examples

Table 71. RACDCERT REKEY examples

Example 1

Operation  User RACFADM has an expiring CERTAUTH certificate labeled 'Local PKI CA' and wants to renew it and rekey the private key. The new, rekeyed certificate will be labeled 'Local PKI CA-2'. The PCI cryptographic coprocessor will be used to generate the new key pair. The size of the new private key will be 1024 bits (RACF default size). After issuing the RACDCERT REKEY command, the user RACFADM will issue the RACDCERT ROLLOVER command to retire and replace the expiring certificate.

Known  User RACFADM has CONTROL access to the IRR.DIGTCERT.REKEY resource in the FACILITY class.

Command

RACDCERT REKEY(LABEL('Local PKI CA'))
CERTAUTH
WITHLABEL('Local PKI CA-2')
PCICC

Output  None.
RACDCERT REMOVE (Remove certificate from key ring)

Purpose
Use the RACDCERT REMOVE command to remove a digital certificate from a key ring.

See "UTF-8 and BMP character restrictions" on page 278 for information about how UTF-8 and BMP characters in certificate names and labels are processed by RACDCERT functions.

Issuing options
The following table identifies the eligible options for issuing the RACDCERT REMOVE command:

<table>
<thead>
<tr>
<th>As a RACF TSO command?</th>
<th>As a RACF operator command?</th>
<th>With command direction?</th>
<th>With automatic command direction?</th>
<th>From the RACF parameter library?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>No</td>
<td>No. (See rules.)</td>
<td>No. (See rules.)</td>
<td>No</td>
</tr>
</tbody>
</table>

Rules:
- The RACDCERT command cannot be directed to a remote system using the AT or ONLYAT keyword.
- The updates made to the RACF database by RACDCERT are eligible for propagation with automatic direction of application updates based on the RRSFDATA profiles AUTODIRECT.target-node.DIGTCERT.APPL and AUTODIRECT.target-node.DIGTRING.APPL, where target-node is the remote node to which the update is to be propagated.

Authorization required
To issue the RACDCERT REMOVE command, you must have the SPECIAL attribute or sufficient authority to the IRR.DIGTCERT.REMOVE resource in the FACILITY class for your intended purpose.

Table 72. Authority required for the RACDCERT REMOVE function

<table>
<thead>
<tr>
<th>IRR.DIGTCERT.REMOVE</th>
<th>Access level</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>READ</td>
<td>Remove a certificate from your own key ring.</td>
<td></td>
</tr>
<tr>
<td>UPDATE</td>
<td>Remove a SITE or CERTAUTH certificate from your own key ring.</td>
<td></td>
</tr>
<tr>
<td>CONTROL</td>
<td>Remove a certificate from another user's key ring.</td>
<td></td>
</tr>
</tbody>
</table>

Activating your changes
If the DIGTCERT or DIGTRING class is RACLISTed, refresh the classes to activate your changes.

Example:
SETROPTS RACLIST(DIGTCERT, DIGTRING) REFRESH

Related commands
- To connect a certificate to a key ring, see RACDCERT CONNECT.
- To list a key ring, see RACDCERT LISTRING.
RACDCERT REMOVE

Syntax

For the key to the symbols used in the command syntax diagrams, see "Syntax of RACF commands and operands" on page 9. The complete syntax of the RACDCERT REMOVE command is:

```
RACDCERT REMOVE([ID(certificate-owner) | SITE | CERTAUTH]

LABEL('label-name')
RING(ring-name)
)
[ ID(ring-owner) ]
```

If you specify more than one RACDCERT function, only the last specified function is processed. Extraneous keywords that are not related to the function being performed are ignored.

If you do not specify a RACDCERT function, LIST is the default function.

For information on issuing this command as a RACF TSO command, refer to Chapter 3, “RACF TSO commands,” on page 15.

Parameters

```
REMOVE(ID(certificate-owner) LABEL('label-name') RING(ring-name))
REMOVE(SITE LABEL('label-name') RING(ring-name))
REMOVE(CERTAUTH LABEL('label-name') RING(ring-name))
```

- **REMOVE** specifies the digital certificate to be removed from the key ring.
- **ID(certificate-owner)** indicates that the certificate being removed is a user certificate, and **certificate-owner** is the user ID associated with this certificate.
- **SITE** indicates that the certificate being removed is a site certificate, and **CERTAUTH** indicates that it is a certificate authority certificate. If **ID**, **SITE** or **CERTAUTH** are not specified, **ID(certificate-owner)** defaults to the key ring owner as specified or defaulted by the **ID(ring-owner)** keyword.
- **LABEL('label-name')** identifies the certificate that is being removed from the key ring. You must specify a label.
- **RING(ring-name)** identifies the key ring from which this certificate is being removed. You must specify a ring name. **Note**: The key ring belongs to the ID specified or defaulted by the **ID(ring-owner)** keyword.
- **ID(ring-owner)** specifies the user ID of the key ring owner. (Only a user ID can have a key ring.) If not specified, the key ring owner defaults to the command issuer’s user ID.

Examples

**Table 73. RACDCERT REMOVE examples**

<table>
<thead>
<tr>
<th>Example 1</th>
<th>Operation</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>User RACFADM wants to remove a SITE certificate with the label Shared Server from the RING01 key ring of server INVSERV.</td>
<td>RACDCERT ID(INVSERV) REMOVE(SITE LABEL('Shared Server') RING(RING01))</td>
</tr>
<tr>
<td>Known</td>
<td>User RACFADM has SPECIAL authority.</td>
<td></td>
</tr>
<tr>
<td>Command</td>
<td>RACDCERT ID(INVSERV) REMOVE(SITE LABEL('Shared Server') RING(RING01))</td>
<td>Output None.</td>
</tr>
</tbody>
</table>
**RACDCERT ROLLOVER (Rollover certificate)**

**Purpose**

Use the RACDCERT ROLLOVER command to supersede one certificate (the source certificate) with another certificate (the target certificate). In general, issue the RACDCERT ROLLOVER command after issuing the RACDCERT REKEY command to supersede an old, expiring certificate with a new rekeyed certificate, and to retire the private key of the expiring certificate. For sample procedures, see "Renewing a certificate with a new private key (rekeying)" in [z/OS Security Server RACF Security Administrator's Guide](#).

Both the source and target certificates are associated with the user ID, CERTAUTH, or SITE as specified on the command. RACDCERT ROLLOVER processing performs the following actions in the specified order:

1. Deletes the private key of the source certificate so that it may not be used again for signing or encryption.

2. Adds the target certificate to any key ring that contains the source certificate and, depending on how the source certificate is connected to the ring, RACDCERT ROLLOVER processing also does one of the following actions:
   - If the source certificate is connected with PERSONAL usage, the source certificate is replaced by the target certificate. In other words, the new certificate is added to the ring and the old one is removed.
   - If the source certificate is connected with CERTAUTH or SITE usage, the target certificate is added to the key ring and the source certificate remains connected. In other words, the new certificate is added to the ring but the old one is not removed.

3. Adds the target certificate to each key ring that contains the source certificate, and depending on how the source certificate is connected to the ring, RACDCERT ROLLOVER processing also does one of the following actions:

4. Copies the serial number base from the source certificate to the target certificate.

Once rollover is complete, the new certificate may be used as if it were the old certificate. The old certificate is retained for historical reasons such as validating signatures on existing certificates, but may no longer be used for any private key operations such as signing other certificates.

**Issuing options**

The following table identifies the eligible options for issuing the RACDCERT ROLLOVER command:
RACDCERT ROLLOVER

<table>
<thead>
<tr>
<th>As a RACF TSO command?</th>
<th>As a RACF operator command?</th>
<th>With command direction?</th>
<th>With automatic command direction?</th>
<th>From the RACF parameter library?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>No</td>
<td>No. (See rules.)</td>
<td>No. (See rules.)</td>
<td>No</td>
</tr>
</tbody>
</table>

Rules:
- The RACDCERT command cannot be directed to a remote system using the AT or ONLYAT keyword.
- The updates made to the RACF database by RACDCERT are eligible for propagation with automatic direction of application updates based on the RRSFDATA profiles AUTODIRECT.target-node.DIGTCERT.APPL and AUTODIRECT.target-node.DIGTRING.APPL, where target-node is the remote node to which the update is to be propagated.

Authorization required

To issue the RACDCERT ROLLOVER command, you must have the SPECIAL attribute or sufficient authority to the IRR.DIGTCERT.ROLLOVER resource in the FACILITY class for your intended purpose.

Table 74. Authority required for the RACDCERT ROLLOVER function

<table>
<thead>
<tr>
<th>IRR.DIGTCERT.ROLLOVER</th>
<th>Access level</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>READ</td>
<td>Rollover your own certificate.</td>
</tr>
<tr>
<td></td>
<td>UPDATE</td>
<td>Rollover another user’s certificate.</td>
</tr>
<tr>
<td></td>
<td>CONTROL</td>
<td>Rollover a SITE or CERTAUTH certificate.</td>
</tr>
</tbody>
</table>

Activating your changes

If the DIGTCERT class is RACLISTed, refresh the class to activate your changes.

Example:

```
SETROPTS RACLIST(DIGTCERT) REFRESH
```

Related commands

- To rekey a expiring certificate, see RACDCERT REKEY.

Syntax

For the key to the symbols used in the command syntax diagrams, see “Syntax of RACF commands and operands” on page 9. The complete syntax of the RACDCERT ROLLOVER command is:

```
RACDCERT ROLLOVER(‘old-label-name’)
    [ ID(‘certificate-owner’) | SITE | CERTAUTH ]
    NEWLABEL(‘new-label-name’)
    [ FORCE ]
```

If you specify more than one RACDCERT function, only the last specified function is processed. Extraneous keywords that are not related to the function being performed are ignored.

If you do not specify a RACDCERT function, LIST is the default function.
RACDCERT ROLLOVER

For information on issuing this command as a RACF TSO command, refer to Chapter 3, “RACF TSO commands,” on page 15.

Parameters

RO Archive LABEL('old-label-name')

Specifies the label of the source certificate to be superseded by the certificate with the 'new-label-name' label.

ID(certificate-owner) | SITE | CERTAUTH

Specifies that both certificates identified by LABEL and NEWLABEL are either user certificates associated with the specified user ID, site certificates, or a certificate-authority certificates. If you do not specify ID, SITE, or CERTAUTH, the default is ID, and certificate-owner defaults to the user ID of the command issuer. If more than one keyword is specified, the last specified keyword is processed and the others are ignored by TSO command parse processing.

NEWLABEL('new-label-name')

Specifies the label of the target certificate for the rollover function. This keyword is required and must identity an existing certificate owned by the specified user ID, SITE, or CERTAUTH.

FORCE

Specifies that RACF should bypass some error checking and unconditionally perform the rollover operation.

If you do not specify FORCE, all of the following conditions must be true or an error message is issued and the command ends:

• The values specified for the LABEL and NEWLABEL keywords must be different.
• Each certificate identified by the LABEL and NEWLABEL keywords must have a private key associated with it.
• The certificate identified by the NEWLABEL keyword was never the target certificate of a previously issued RACDCERT ROLLOVER command and was never used to sign other certificates.

If you specify FORCE, these conditions are not checked. If you specify FORCE and inadvertently specify the same certificate with both the LABEL and NEWLABEL keywords, the private key of this certificate is deleted.

Examples

Table 75. RACDCERT ROLLOVER examples

Example 1

Operation  User RACFADM has an expiring CERTAUTH certificate labeled 'Local PKI CA' and wants to retire it and replace it with a new, rekeyed certificate labeled 'Local PKI CA-2'.

Known  User RACFADM has CONTROL access to the IRR.DIGTCERT.ROLLOVER resource in the FACILITY class.

Command  RACDCERT ROLLOVER(LABEL('Local PKI CA'))

CERTAUTH

NEWLABEL('Local PKI CA-2')

Output  None.
RACDCERT UNBIND

RACDCERT UNBIND (Unbind certificate from token)

Purpose

Removes a digital certificate from the specified z/OS PKCS #11 token.

See “UTF-8 and BMP character restrictions” on page 278 for information about how UTF-8 and BMP characters in certificate names and labels are processed by RACDCERT functions.

Issuing options

The following table identifies the eligible options for issuing the RACDCERT UNBIND command:

<table>
<thead>
<tr>
<th>As a RACF TSO command?</th>
<th>As a RACF operator command?</th>
<th>With command direction?</th>
<th>With automatic command direction?</th>
<th>From the RACF parameter library?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>No</td>
<td>No. (See rules.)</td>
<td>No. (See rules.)</td>
<td>No</td>
</tr>
</tbody>
</table>

Rules:

- The RACDCERT command cannot be directed to a remote system using the AT or ONLYAT keyword.
- The updates made to the RACF database by RACDCERT are eligible for propagation with automatic direction of application updates based on the RRSFDATA profiles AUTODIRECT.target-node.DIGTCERT.APPL and AUTODIRECT.target-node.DIGTRING.APPL, where target-node is the remote node to which the update is to be propagated.

Authorization required

Authorization to delete z/OS PKCS #11 tokens is controlled by ICSF based on profiles in the CRYPTOZ class. (No authority in the FACILITY class is required.) If the command issuer does not have authority to remove the certificate from the specified token as determined by ICSF, the command stops and an error message is displayed.

If the CSFSERV class is active, the command issuer might also need sufficient authority in the CSFSERV class.

For authorization details about the CRYPTOZ and CSFSERV classes, see Z/OS Cryptographic Services ICSF Administrator’s Guide.

Related commands

- To bind a certificate to a token, see RACDCERT BIND.
- To list a token, see RACDCERT LISTTOKEN.

Syntax

For the key to the symbols used in the command syntax diagrams, see “Syntax of RACF commands and operands” on page 9. The complete syntax of the RACDCERT UNBIND command is:
RACDCERT UNBIND

RACDCERT UNBIND(TOKEN(token-name)
   { [ ID(certificate-owner) | SITE | CERTAUTH ] LABEL('label-name')
   | SEQNUM(sequence-number) }
)

[ FORCE ]

Note: Unless specified as a subkeyword of the UNBIND parameter, the
   ID(certificate-owner) | SITE | CERTAUTH parameter is ignored for the
   RACDCERT UNBIND function.

If you specify more than one RACDCERT function, only the last specified function is
processed. Extraneous keywords that are not related to the function being
performed are ignored.

If you do not specify a RACDCERT function, LIST is the default function.

For information on issuing this command as a RACF TSO command, refer to

Parameters

UNBIND(TOKEN(token-name) ID(certificate-owner) LABEL('label-name'))
UNBIND(TOKEN(token-name) SITE LABEL('label-name'))
UNBIND(TOKEN(token-name) CERTAUTH LABEL('label-name'))
UNBIND(TOKEN(token-name) SEQNUM(sequence-number))
  You must uniquely identify the certificate to be removed in one of the following
  ways:
  • By its RACF label name (if defined to RACF), and optionally identify it as a
    user, SITE or CERTAUTH certificate. (The certificate must be defined to
    RACF when you specify the label name.)
  • By its sequence number within the token. (The certificate need not be
    defined to RACF when you specify the sequence number.)

TOKEN(token-name)
  Specifies the name of the token from which the certificate is being removed.
  When specifying the UNBIND operand, you must specify the TOKEN operand.

ID(certificate-owner) | SITE | CERTAUTH
  Specifies that the certificate to remove from the token is either a user certificate
  associated with the specified user ID, a site certificate, or a certificate-authority
  certificate. If you do not specify ID, SITE, or CERTAUTH, the default is ID, and
  certificate-owner defaults to the user ID of the command issuer. If more than
  one keyword is specified, the last specified keyword is processed and the
  others are ignored by TSO command parse processing.

LABEL('label-name')
  Specifies the certificate to remove from the token. LABEL is mutually exclusive
  with SEQNUM.

SEQNUM(sequence-number)
  Specifies the sequence number of the certificate to remove from the token. If
  the certificate (or its associated private key, if any) is not currently defined to
  RACF, you must also specify FORCE or else an error message is issued and
  the command ends. (This error prevents you from inadvertently deleting a
  certificate that is not defined to RACF.) SEQNUM is mutually exclusive with
  LABEL.
FORCE
Specifies that RACF should bypass some error checking and unconditionally perform the unbind operation.

If you do not specify FORCE, the following condition must be true or an error message is issued and the command ends:

- The certificate (or its associated private key, if any) must be currently defined to RACF.

If you specify FORCE, this condition is not checked. If you specify FORCE and inadvertently specify a sequence number for a certificate not defined to RACF, the certificate (or its associated private key, if any) is deleted.

Examples

Table 76. RACDCERT UNBIND examples

Example 1

*Operation*  User NETB0Y wishes to remove a digital certificate labeled My temp certificate from the z/OS PKCS #11 token named NETB0Y.TKN1. The certificate does not currently reside in RACF.

*Known*  User NETB0Y has CONTROL authority to the discrete profile named USER.NETB0Y.TKN1 in the CRYPTOZ class. Using RACDCERT LISTTOKEN, user NETB0Y determined the sequence number of the certificate to be removed is 3.

*Command*  RACDCERT UNBIND(TOKEN(NETB0Y.TKN1) SEQNUM(3)) FORCE

*Output*  None.
RACLINK (Administer user ID associations)

Purpose

Use the RACLINK command to:

- Define, approve, and delete (undefine) an established or pending user ID association
- List information related to a user ID association
- Establish password synchronization between user IDs

Notes:

1. When the RACLINK command is issued from ISPF, the TSO command buffer (including password data) is written to the ISPLOG data set. As a result, you should not issue this command from ISPF or you must control the ISPLOG data set carefully.
2. If the RACLINK command is issued as a RACF operator command, the command and the password data is written to the system log. Therefore, use of RACLINK as a RACF operator command should either be controlled or you should issue the command as a TSO command.

Issuing options

The following table identifies the eligible options for issuing the RACLINK command:

<table>
<thead>
<tr>
<th>As a RACF TSO command?</th>
<th>As a RACF operator command?</th>
<th>With command direction?</th>
<th>With automatic command direction?</th>
<th>From the RACF parameter library?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
</tbody>
</table>

For information on issuing this command as a RACF TSO command, refer to Chapter 3, “RACF TSO commands,” on page 15.

For information on issuing this command as a RACF operator command, refer to Chapter 4, “RACF operator commands,” on page 21.

You must be logged on to the console to issue this command as a RACF operator command.

Related commands

- To add a user profile, see ADDUSER (Add user profile)
- To display information from a user profile, see LISTUSER (List user profile)
- To change a user profile, see ALTUSER (Alter user profile)
- To delete a user profile, see DELUSER (Delete user profile)
- To obtain a list of user profiles, see SEARCH (Search RACF database)

Authorization required

You have the authority to issue the RACLINK command for your own user ID.

To issue the RACLINK DEFINE command you must also have sufficient authority to the proper profiles in the RRSFDATA class. For RACLINK DEFINE, this is the first security check performed. For more information, see z/OS Security Server RACF Security Administrator’s Guide.
You can issue the RACLINK command for a user ID other than your own if you have the following authority over the user ID specified on the ID keyword:

- You have the SPECIAL attribute.
- The profile is within the scope of a group in which you have the group-SPECIAL attribute.
- You are the profile owner.

When the DEFINE keyword is specified and the command issuer has sufficient authority to perform the RACLINK command for the user ID, the user ID association is implicitly approved if:

- A valid password is supplied for the user ID specified on the DEFINE keyword.
- The command issuer has one of the following authorities over the user ID specified on the DEFINE keyword:
  - The command issuer has the SPECIAL attribute.
  - The profile is within the scope of a group in which the command issuer has the group-SPECIAL attribute.
  - The command issuer is the owner of the profile.
- The command issuer has an association with a user ID on the node specified on the DEFINE keyword. That association must be either a PEER association or a MANAGED association with the command issuer as the manager. The user ID with which the command issuer has the association must have one of the following authorities over the user ID specified on the DEFINE keyword:
  - The command issuer has the SPECIAL attribute.
  - It is within the scope of a group that has the group-SPECIAL attribute.
  - It is the owner of the profile.

When issuing the RACLINK command as a RACF operator command, you might require sufficient authority to the proper resource in the OPERCMDS class. See z/OS Security Server RACF Security Administrator's Guide for further information.

Syntax

For the key to the symbols used in the command syntax diagrams, see “Syntax of RACF commands and operands” on page 9. The complete syntax of the RACLINK command is:

```plaintext
[subsystem-prefix]RACLINK
    [ ID(userid1 ...)]
    [ LIST ([node | *].[userid2 | *] ...) ]
    [ DEFINE ([node].userid2/password] ... ]
    [ MANAGED | PEER (NOPWSYNC | PWSYNC) ]
    [ UNDEFINE ([node].userid2 ... ] ]
    [ APPROVE ([node].userid2 ... ] ]
```

For information on issuing this command as a RACF TSO command, refer to Chapter 3, “RACF TSO commands,” on page 15.

For information on issuing this command as a RACF operator command, refer to Chapter 4, “RACF operator commands,” on page 21.
Parameters

subsystem-prefix

Specifies that the RACF subsystem is the processing environment of the command. The subsystem prefix can be either the installation-defined prefix for RACF (1–8 characters) or, if no prefix has been defined, the RACF subsystem name followed by a blank. If the command prefix was registered with CPF, you can use the MVS command D OPDATA to display it or you can contact your RACF security administrator.

Only specify the subsystem prefix when issuing this command as a RACF operator command. The subsystem prefix is required when issuing RACF operator commands.

ID(userid1 ...)

Specifies the user for whom the RACLINK operation is to be performed. Specify one or more user IDs on the RRSF node from which the command is issued.

If this operand is not specified, the command defaults to the user issuing the command at the node where the command is issued.

LIST([node][.|*].[userid2 |.|*] ...)

Specifies that a list of associations for node.userid2 is to be displayed. If multiple user IDs are specified, then multiple lists are displayed, one for each user ID specified.

RACLINK LIST (*.* is the default. RACLINK LIST (*.* lists all user ID associations for the specified user ID or the issuer’s user ID if the ID keyword is not specified.

If the node name is not specified, the default is the local node.

The node names you specify must have been defined as RRSF nodes with the TARGET command.

The following information is displayed for each user ID association:

- User ID association type
  - Peer association
  - Managed association (including whether the specified user ID is the managed user ID or the managing user ID)

- Password synchronization status
  - YES (password synchronization is active)
  - NO (password synchronization is inactive)
  - N/A (password synchronization is not applicable to a managed association)

- User ID association status
  - PENDING APPROVAL BY userid (waiting for userid to approve or reject the user ID association)
  - ESTABLISHED (the user ID association has been approved)
  - SYSTEM ERROR (an unexpected error occurred on the target node that prevented the user ID association from being completed) The user ID association should be deleted and then defined again.

DEFINE([node].userid2[/password] ...)

Specifies that a user ID association is to be formed between userid1 at the node where the command was issued, and userid2 at node. If you specify more than one node.userid2 operand, an association is established between userid1
and each node.userid2 specified. A user ID association enables RACF users to utilize command direction and password synchronization.

To issue the RACLINK DEFINE command, you need READ access to the following profile in the RRSFDATA class:

- RA CLINK.DEFINE.node

The RRSFDATA class must be active.

When the DEFINE keyword is specified and the command issuer has sufficient authority to perform the RACLINK command for the user ID, the user ID association is implicitly approved if:

- A valid password is supplied for node.userid2 on the DEFINE keyword.
- The command issuer has one of the following authorities over userid2 on the DEFINE keyword:
  - The command issuer has the SPECIAL attribute.
  - The profile is within the scope of a group in which the command issuer has the group-SPECIAL attribute.
  - The command issuer is the owner of the profile.
- The command issuer has an association with a user ID on the node specified on the DEFINE keyword. That association must be either a PEER association or a MANAGED association with the command issuer as the manager. The user ID with which the command issuer has the association must have one of the following authorities over userid2 on the DEFINE keyword:
  - The command issuer has the SPECIAL attribute.
  - It is within the scope of a group that has the group-SPECIAL attribute.
  - It is the owner of the profile.

Otherwise, a user ID association requires explicit approval by node.userid2 with the RACLINK APPROVE command.

Although it is possible for the command issuer to have more than 50 associated user IDs on the target node, only the first 50 are used for authority checking. RACLINK issues a message if more than 50 user ID associations exist for the command issuer.

An association is PENDING until node.userid2 either approves the association with a RACLINK APPROVE command or rejects the association with a RACLINK UNDEFINE command.

Notes:

1. Under certain circumstances, RACLINK DEFINE(node.userid) requests can be issued by two users. If both requests are consistent, RACF treats this as an implicit approval. The entry is marked established in the target user IDs profile. An entry is considered consistent if the association type (PEER(PWSYNC) or PEER(NOPWSYNC)) is the same. If the request is not consistent (for example, differing PEER definitions or both users requesting a MANAGED association), RACF fails the request and the entries remain in a pending state. In order to correct this situation, the user(s) need to undefine and redefine the user ID associations.

2. When creating a user ID association with a revoked user ID:
   - If a RACLINK DEFINE command is coded without the password operand and the target user ID is a revoked user, the results vary depending on the authority of the command issuer and the user ID associations of the command issuer. When:
The user ID association with the revoked user ID is created and the status displayed by a RACLINK LIST command is "ESTABLISHED" when one of the following is true:

- The command issuer has sufficient authority (SPECIAL, group-SPECIAL, or owner) over the target user ID or
- The command issuer has a PEER association or is the manager of a MANAGED association with a user ID on the target node and the associated user ID has sufficient authority over the target user ID.

3. If a RACLINK DEFINE command is coded without the password operand, the target user ID is a revoked user ID, and the command issuer does not have sufficient authority (SPECIAL, group-SPECIAL, or owner) over the target user ID, the user ID association is created and the status displayed by a RACLINK LIST command is "PENDING APPROVAL BY userid2".

4. If a RACLINK DEFINE command is coded with the password operand and the target user is a revoked user, the user ID association is not established and the status displayed by a RACLINK LIST is "SYSTEM ERROR".

The type of association you want to establish is specified with one of the following:

**MANAGED**

- Specifies a managed association.

  A managed association does not provide password synchronization. A managed association allows commands to be directed from the managing user ID to the managed user ID (that is, from userid1 to node.userid2).

  A managed association does not allow commands to be directed from the managed user ID to the managing user ID (that is, node.userid2 cannot direct commands to userid1).

**PEER(NOPWSYNC)**

- Specifies a peer association without password synchronization.

  Either user ID in a peer association can direct commands to the other user ID in the association.

  If no association type is specified, PEER(NOPWSYNC) is the default.

**PEER(PWSYNC)**

- Specifies a peer association with password synchronization.

  Either user ID in a peer association can direct commands to the other user ID in the association.

  If either user in the association changes their password, the password is automatically changed for the other user in the association.

  READ access to the RACLINK.PWSYNC.node resource is required to use the RACLINK command to define a peer association with the PWSYNC attribute. READ access to the PWSYNC resource is required to synchronize the passwords when one of the associated users changes their password.

  If the RRSFDATA class is not active, you cannot define an association with the PWSYNC attribute, or synchronize passwords.

**UNDEFINE([node].userid2 ...)**

- Specifies that a user ID association is ended between userid2 on node and userid1 on the node where the command is processed. Either member of an association can end an association.

Chapter 5. RACF command syntax  389
RACLINK

If a user ID has attempted to establish an association with your user ID which requires approval, and you do not want to approve it, use the UNDEFINE keyword to reject the pending association.

APPROVE([node1].userid1 ...)]

Specifies that userid2 on node2 approves of a pending association between userid2 at node2 and userid1 at node1. node1 is the node where the RACLINK DEFINE was issued, and node2 is the node where userid2 issues the command.

Examples

Table 77. RACLINK examples

Example 1

| Operation | The security administrator wants to know what, if any, associations user DENICE has with user BETH. |
| Known | The security administrator wants to issue the command as a RACF TSO command. |
| Command | RACLINK ID(DENICE) LIST(*.BETH) |
| Defaults | None. |
| Output | See Figure 48. |

Example 2

| Operation | User DENICE wants to define password synchronization between all of her MVS user IDs; DENICE at NODE1, DENICE at NODE2, and DENICE at NODE3. |
| Known | DENICE wants to issue the command as a RACF TSO command. DENICE has the authority to issue the RACLINK command for her own user IDs and has the authority to establish password synchronization for her own user IDs. The command is to be issued from DENICE at NODE1. |
| Command | RACLINK DEFINE(NODE2.DENICE/passw2 NODE3.DENICE/passw3) PEER(PWSYNC) |
| Defaults | None. |
| Results | DENICE at NODE1 receives the following messages: |
| | IRRT032I RACLINK command to associate user ID DENICE with NODE2.DENICE is pending approval. |
| | IRRT032I RACLINK command to associate user ID DENICE with NODE3.DENICE is pending approval. |
| | IRRP097I Peer association with DENICE at node NODE2 has been approved. |
| | IRRP097I Peer association with DENICE at node NODE3 has been approved. |

When user DENICE changes her password on one of her MVS user IDs, the new password propagates to take affect on her other user IDs. The password is checked for validity only on the node where user DENICE issues the command to change her password, not at any of the other nodes.
Table 77. RACLINK examples (continued)

**Example 3**

**Operation** User BETH wants to define a MANAGED user ID association where BETH is the managing user ID and DENICE is the managed user ID.

**Known** User BETH:
- wants to issue the command as a RACF TSO command,
- does not know the password for user DENICE, and
- has the authority to issue the RACLINK command for her own user ID.

**Command** RACLINK DEFINE(NODE1.DENICE) MANAGED

**Defaults** None.

**Results** User BETH receives the following message:

IRRT032I RACLINK command to associate user ID BETH with NODE1.DENICE is pending approval.

User DENICE receives the following message:

IRRP094I Managed association with DENICE at node NODE1 issued by BETH waiting for your approval.

The association remains pending until DENICE at NODE1 either approves the association with a RACLINK APPROVE command or rejects the association with a RACLINK UNDEFINE command.

<table>
<thead>
<tr>
<th>Association Type</th>
<th>Node.userid</th>
<th>Password</th>
<th>Sync</th>
<th>Association Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>PEER OF</td>
<td>NODE1.BETH</td>
<td>YES</td>
<td></td>
<td>ESTABLISHED</td>
</tr>
<tr>
<td>MANAGED BY</td>
<td>NODE2.BETH</td>
<td>N/A</td>
<td></td>
<td>PENDING APPROVAL BY DENICE</td>
</tr>
<tr>
<td>PEER OF</td>
<td>NODE3.BETH</td>
<td>NO</td>
<td></td>
<td>PENDING APPROVAL BY BETH</td>
</tr>
</tbody>
</table>

*Figure 48. Example 1: Output for the RACLINK LIST Command*
**RACMAP (Create, delete, or list a distributed identity filter)**

**Purpose**

Use the RACMAP command to create, delete, and list a distributed identity filter. A distributed identity filter is a mapping association between a RACF user ID and one or more distributed user identities. The filter consists of all or selected components of a distributed-identity user name and the distributed-identity registry name.

When you use the RACMAP command to add a distributed identity filter, RACF creates a general resource profile in the IDIDMAP class. RACF uses distributed identity filters to determine the RACF user ID of a user who attempts to access the system using a distributed identity.

RACF accepts distributed user information from authorized applications that issue the RACROUTE REQUEST=VERIFY request or the initACEE callable service (IRRSIA00), and determines the RACF user ID of the distributed user by matching the distributed-identity user name and registry name with the filters in IDIDMAP class profiles.

For information about how to use a distributed identity filter to map distributed identities to a RACF user ID, see "Distributed identity filters" in *z/OS Security Server RACF Security Administrator’s Guide*.

**Issuing options**

The following table identifies the eligible options for issuing the RACMAP command:

<table>
<thead>
<tr>
<th>As a RACF TSO command?</th>
<th>As a RACF operator command?</th>
<th>With command direction?</th>
<th>With automatic command direction?</th>
<th>From the RACF parameter library?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>No</td>
<td>No. (See rules.)</td>
<td>No. (See rules.)</td>
<td>No</td>
</tr>
</tbody>
</table>

**Rules:**

- The RACMAP command cannot be directed to a remote system using the AT or ONLYAT keyword.
- Updates made to the RACF database by RACMAP are eligible for propagation with automatic direction of application updates based on the RRSFDATA profile named AUTODIRECT.target-node.IDIDMAP.APPL, where target-node is the remote node to which the update is to be propagated.

For information on issuing this command as a RACF TSO command, see Chapter 3, “RACF TSO commands,” on page 15.

**Related commands**

None.

The MAP, DELMAP and LISTMAP functions of the RACMAP command are unrelated to the MAP, DELMAP and LISTMAP functions of the RACDCERT command.

**Authorization required**

To issue the RACMAP command, you must have SPECIAL authority or sufficient authority to the IRR.IDIDMAP.function resource in the FACILITY class, where function is MAP, DELMAP, or LISTMAP.
### Activating your changes

To activate your changes, you must activate and RACLIST the IDIDMAP class.

When you create a distributed identity filter for the first time, issue the following command.

**Example:**

```plaintext
SETROPTS CLASSACT(IDIDMAP) RACLIST(IDIDMAP)
```

If the IDIDMAP class is already active and RACLISTed, refresh the class to activate your changes.

**Example:**

```plaintext
SETROPTS RACLIST(IDIDMAP) REFRESH
```

### Syntax

For the key to the symbols used in the command syntax diagrams, see "Syntax of RACF commands and operands" on page 9. The complete syntax of the RACMAP command is:

```plaintext
RACMAP
  [ ID(userid) ]
  MAP
    USERDIDFILTER(NAME('distributed-identity-user-name' | '*'))
    REGISTRY(NAME('distributed-identity-registry-name' | '*'))
    [ WITHLABEL('label-name') ]
  | DELMAP
    [ (LABEL('label-name')) ]
  | LISTMAP
    [ (LABEL('label-name')) ]
```

For information on issuing this command as a RACF TSO command, refer to Chapter 3, “RACF TSO commands,” on page 15.

### Parameters

**ID(userid)**

Specifies the RACF user ID mapped by the distributed identity filter. The user ID must already be defined to RACF. If you do not specify ID, the default value is the RACF user ID of the command issuer.

**MAP**

Specifies the MAP function of the RACMAP command. Use the MAP function to create a distributed identity filter that maps a user's distributed identity to a RACF user ID. The MAP function creates a profile in the IDIDMAP class for each filter you create.
For information about how to use a distributed identity filter to map distributed identities to a RACF user ID, see z/OS Security Server RACF Security Administrator's Guide.

When you specify the MAP function, you must specify name values for USERDIDFILTER and REGISTRY.

**USERDIDFILTER(NAME('distributed-identity-user-name' | '*'))**

Specifies the significant portion of the distributed-identity user name. RACF uses the user name as part of the distributed identity filter to map a distributed identity to a RACF user ID.

Specify the user name value enclosed in single quotation marks. If a single quotation mark is intended to be part of the name, specify two single quotation marks together for each single quotation mark in the name, and enclose the entire name value in single quotation marks.

The maximum length for a user name is 246 bytes.

**Restriction for names containing multibyte characters:** Because RACF converts the name value you specify from EBCDIC to UTF-8 format prior to storing it in the RACF database, if your value contains multibyte characters, the resulting UTF-8 value might be longer than 246 bytes. If this occurs, the command fails and message IRRW213I is issued.

In general, the user name can contain blank and mixed-case characters. Any leading and trailing blank characters are removed from the value before it is stored in the IDIDMAP profile.

You cannot specify the name value as a hexadecimal character string.

**Format of the user name value:** Specify the user name value in any of the following three formats:

1. As a single asterisk (X'5C') to indicate that any user name matches this filter.
2. As a simple character string, such as a user ID defined in a non-LDAP registry.
   - Typically, special characters do not appear in user names stored within a registry. However, if you need to specify a user name value that includes certain characters, they must be preceded by the backslash (\) escape character.
   - These characters include the plus sign (+), semicolon (;), comma (,), quotation mark ("), backslash (\), less than symbol (<), greater than symbol (>), and the equal sign (=).
3. As a character string that represents an X.500 distinguished name (DN).
   - A DN consists of one or more relative distinguished names (RDNs).
   - Each RDN consists of an attribute type and attribute value, separated by an equal sign (=). RDNs are separated by a comma (,).
   - When you use mixed-case characters to specify the user name as a DN, the RACMAP command translates the attribute types to uppercase characters, and preserves the mixed-case characters of the attribute value.
   - The RACMAP command performs no validity checking of the X.500 names you specify.

**Rules when defining the user name as a distinguished name (DN):**
Specify the user name value in its canonical form, as it is defined within the registry, with any special characters preceded by the backslash (\) escape character. You must specify the RDNs in their correct sequence.

For example, for users of WebSphere® Application Server applications, the canonical form of the user name must match the value returned by the WSCredential interface method called getUniqueSecurityName().

- Typically, special characters do not appear in user names stored within a registry. However, if you need to specify a user name value that includes certain characters, including LDAP special characters, they must be preceded by the backslash (\) escape character.

These characters include the plus sign (+), semicolon (;), comma (,), quotation mark ("), backslash (\), less than symbol (<), greater than symbol (>), and the equal sign (=).

**Exception:** Do not escape the equal sign (=) or comma (,) when you specify them as delimiters of an RDN.

- Do not specify blank characters immediately preceding or following the equal sign (=) when using the equal sign as a delimiter of an RDN.

**Examples:**

```plaintext
USERIDFILTER(NAME('DENICE'))
USERIDFILTER(NAME('UID=GUSKI,OU=Tools Development,O=IBM,C=US'))
USERIDFILTER(NAME('Rich''s ID'))
USERIDFILTER(NAME('Dev\+Test219'))
USERIDFILTER(NAME('*'))
```

The USERIDFILTER operand is required for the MAP function and ignored if the RACMAP function is not MAP.

**REGISTRY(NAME(’distributed-identity-registry-name’ ’''))**

Specifies the registry that contains the distributed-identity user name. RACF uses the registry name as part of the distributed identity filter to map a distributed identity to a RACF user ID.

Specify the registry name value enclosed in single quotation marks. If a single quotation mark is intended to be part of the name, specify two single quotation marks together for each single quotation mark in the name, and enclose the entire name value in single quotation marks.

You can specify a single asterisk (X'5C') as the registry name to indicate that any distributed-identity registry name matches this filter.

The maximum length for a registry name is 255 bytes.

**Restriction for names containing multibyte characters:** Because RACF converts the name value you specify from EBCDIC to UTF-8 format prior to storing it in the RACF database, if your value contains multibyte characters, the resulting UTF-8 value might be longer than 255 bytes. If this occurs, the command fails and message IRRW213I is issued.

The registry name can contain blank and mixed-case characters. Any leading and trailing blank characters are removed from the value before it is stored in the IDIDMAP profile.

You cannot specify the name value as a hexadecimal character string.
For users of WebSphere Application Server applications, the registry name must match the value returned by the WSCredential interface method called getRealmName().

**Defining registry names for LDAP servers:** When the user’s distributed identity is based on an LDAP registry, specify the distributed-identity-registry-name value as the URL of the LDAP server where the user is defined. The URL is defined with a listen option in the ds.conf configuration file of the LDAP server, or overridden using the -l command-line parameter when the LDAP server is started.

For information about LDAP URLs, see [IBM Tivoli Directory Server Administration and Use for z/OS](https://www.ibm.com/support/knowledgecenter/SSEKQY_6.3.0/com.ibm.zos.r61.doc/z/dsadm/training/ldapurls.html).

**Examples:**

REGISTRY(NAME('ldaps://us.richradioham.com'))
REGISTRY(NAME('ldap://12.34.56.78:389'))

The RACMAP command performs no validity checking of the registry names you specify.

The REGISTRY operand is required for the MAP function and ignored if the RACMAP function is not MAP.

**WITHLABEL('label-name')**

Specifies the label assigned to this distributed identity filter. If specified, the label must be unique to the RACF user ID associated with this filter.

If WITHLABEL is not specified, RACF generates a label for the filter in the form of LABELnnnnnnnn, where nnnnnnnn is the first integer value, starting at 00000001 that generates a unique label name.

Up to 32 characters can be specified for label-name. The label name can contain blank and mixed-case characters. Any leading and trailing blank characters are removed from the value before it is stored in the IDIDMAP profile.

Specify the label name value enclosed in single quotation marks. If a single quotation mark is intended to be part of the label name, specify two single quotation marks together for each single quotation mark in the name, and enclose the entire label name in single quotation marks.

**DELMAP**

Specifies the DELMAP function of the RACMAP command. Use the DELMAP function to delete a distributed identity filter for the specified RACF user ID. The DELMAP function deletes the profile in the IDIDMAP class that contains the specified filter.

**Rule:** If the specified RACF user ID is associated with more than one filter, you must specify label name.

**LABEL('label-name')**

Specifies the label of the filter to delete for the specified RACF user ID.

**Performance consideration:** When you issue the RACMAP DELMAP command specifying both filter label and a user ID for which no user profile exists, RACF searches all profiles in the IDIDMAP class to locate and delete all distributed identity filters that match. This search might take an extended period of time.
LISTMAP
Specifies the LISTMAP function of the RACMAP command. Use the LISTMAP function to list information about a distributed identity filter for the specified RACF user ID.

**Rule:** If the specified RACF user ID is associated with more than one filter, you must specify label name.

**LABEL('label-name')**
Specifies the label name of the distributed identity filter to list for the specified RACF user ID.

**Tip:** Omit LABEL to list all filters associated with the specified RACF user ID.

If the filter cannot be listed because the IDIDMAP profile containing it is missing or incomplete, the following error text appears in the LISTMAP output:

Filter with label label-name not found.

**Guideline:** When this error text appears in the LISTMAP output, issue a RACMAP DELMAP command specifying this label name to remove residual filter information from the user's profile.

A missing or incomplete IDIDMAP profile might result if a previous RACMAP MAP command failed to complete due to a system failure or early termination by the issuer. If the filter or IDIDMAP profile were not created before the failure, the resulting user profile might contain residual filter information indicating that the RACF user ID is associated with a filter.

If you do not specify the MAP or DELMAP function, LISTMAP is the default function.

**Examples**

<table>
<thead>
<tr>
<th>Table 79. RACMAP examples</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example 1</strong></td>
</tr>
<tr>
<td><strong>Operation</strong></td>
</tr>
<tr>
<td><strong>Known</strong></td>
</tr>
<tr>
<td><strong>Command</strong></td>
</tr>
<tr>
<td><strong>Defaults</strong></td>
</tr>
<tr>
<td><strong>Output</strong></td>
</tr>
</tbody>
</table>

| **Example 2**             |
| **Operation** | The security administrator wants to add a distributed identity filter that specifies the distributed user's name using selected RDNs of the user's X.500 distinguished name. |
| **Known** | The security administrator has the SPECIAL attribute. |
| **Command** | RACMAP ID(ACCTUSER) MAP USERIDIFILTER(NAME('OU=Accounting,O=BobsMart,C=US')) REGISTRY(NAME('ldaps://us.bobsmarturl.com')) WITHLABEL('Accounting office workers') |
| **Defaults** | None. |
| **Output** | None. For a listing of the RACMAP LISTMAP command for this distributed identity filter, see Figure 50 |
**Table 79. RACMAP examples (continued)**

**Example 3**

**Operation**
The security administrator wants to add a distributed identity filter that specifies
the distributed user's name as a non-LDAP user name.

**Known**
The security administrator has the SPECIAL attribute.

**Command**
```racs
RACMAP ID(DENICE) MAP
USERIDFILTER(NAME('DENICE'))
REGISTRY(NAME('Registry01'))
WITHLABEL('Filter for Denice from Registry01')
```

**Defaults** None.

**Output** None. For a listing of the RACMAP LISTMAP command for this distributed
identity filter, see [Figure 51](#).

**Example 4**

**Operation**
The security administrator wants to delete the distributed identity filter labeled
Filter for Denice from Registry01 for the RACF user ID DENICE.

**Known**
The security administrator has the SPECIAL attribute.

**Command**
```racs
RACMAP ID(DENICE) DELMAP(LABEL('Filter for Denice from Registry01'))
```

**Defaults** None.

**Output** None.

---

```racs
RACMAP ID(RLCOOK) LISTMAP
Mapping information for user RLCOOK:
Label: Accounting boss
Distributed Identity User Name Filter:
>UID=BobC,CN=Bob Cook,OU=Accounting,O=BobsMart,C=US<
Registry name:
>ldaps://us.bobsmarturl.com<
```

*Figure 49. Example 1: Output for the RACMAP LISTMAP command*

```racs
RACMAP ID(ACCTUSER) LISTMAP
Mapping information for user ACCTUSER:
Label: Accounting office workers
Distributed Identity User Name Filter:
>OU=Accounting,O=BobsMart,C=US<
Registry name:
>ldaps://us.bobsmarturl.com<
```

*Figure 50. Example 2: Output for the RACMAP LISTMAP command*

```racs
RACMAP ID(DENICE) LISTMAP
Mapping information for user DENICE:
Label: Filter for Denice from Registry01
Distributed Identity User Name Filter:
>DENICE<
Registry name:
>Registry01<
```

*Figure 51. Example 3: Output for the RACMAP LISTMAP command*
RACPRIV (Set write-down privileges)

Purpose

Use the RACPRIV command to allow users, who are authorized to the profile IRR.WRITEDOWN.BYUSER in the FACILITY class, to set, reset, and query the setting of the write-down privilege that they are running within their address space. This command ends with an error message if write-down by user is not active on the system.

To activate write-down by user, the profile IRR.WRITEDOWN.BYUSER must be defined in the FACILITY class, the FACILITY class must be active and RACLISTed, and the SETR MLS option must be active.

Issuing options

The following table identifies the eligible options for issuing the RACPRIV command:

<table>
<thead>
<tr>
<th>As a RACF TSO command?</th>
<th>As a RACF operator command?</th>
<th>With command direction?</th>
<th>With automatic command direction?</th>
<th>From the RACF parameter library?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
</tbody>
</table>

For information on issuing this command as a RACF TSO command, refer to Chapter 3, “RACF TSO commands,” on page 15.

Authorization required

To issue the RACPRIV command, the user must have at least READ access to IRR.WRITEDOWN.BYUSER.

Syntax

For the key to the symbols used in the command syntax diagrams, see Syntax of RACF commands and operands on page 9. The complete syntax of the RACPRIV command is:

```
RACPRIV
   [ WRITEDOWN [( ACTIVE | INACTIVE | RESET )] ]
```

For information on issuing this command as a RACF TSO command, refer to Chapter 3, “RACF TSO commands,” on page 15.

Parameters

**WRITEDOWN**

Specifies the user’s current write-down mode. If WRITEDOWN is specified without a value, or if RACPRIV is issued without any keywords, the current write-down mode is listed. The write-down privilege can only be set, reset, or listed if the FACILITY class is active and RACLISTed, the profile IRR.WRITEDOWN.BYUSER exists in the FACILITY class in the RACLISTed profiles, the SETR MLS option is active, and the user issuing the command has authority to IRR.WRITEDOWN.BYUSER.
**RACPRIV**

**ACTIVE**
Specifies that the user’s write-down mode be set on.

**INACTIVE**
Specifies that the user’s write-down mode be set off.

**RESET**
Specifies that the user’s write-down mode be reset to the user’s installation defined default.

**Examples**

*Table 80. RACPRIV examples*

**Example 1**

*Operation*
User DEBBIE1 wants to know her current write-down setting.

*Known*
- DEBBIE1 has READ access to IRR.WRITEDOWN.BYUSER.
- The FACILITY class is active and RACLITEd.
- The profile IRR.WRITEDOWN.BYUSER exists in the FACILITY class in the RACLITEd profiles.
- The SETROPTS MLS option is active.

*Command*
RACPRIV

*Defaults*
None.

*Output*
User DEBBIE1 may receive the following message:
WRITEDOWN is not currently active for this user

**Example 2**

*Operation*
User DEBBIE1 wants to set her write-down setting to active.

*Known*
- DEBBIE1 has READ access to IRR.WRITEDOWN.BYUSER.
- The FACILITY class is active and RACLITEd.
- The profile IRR.WRITEDOWN.BYUSER exists in the FACILITY class in the RACLITEd profiles.
- The SETROPTS MLS option is active.

*Command*
RACPRIV WRITEDOWN(ACTIVE)

*Defaults*
None.

*Output*
User DEBBIE1 will receive the following message:
WRITEDOWN is currently active for this user
Use the RALTER command to:

- Alter the profile for one or more resources belonging to classes defined in the class descriptor table. Using RALTER to modify an automatic TAPEVOL profile (a profile RACF creates automatically as part of protecting a tape data set) makes that TAPEVOL profile nonautomatic. For more information about TAPEVOL profiles, see z/OS Security Server RACF Security Administrator’s Guide.
- Change the global access checking table
- Change the attributes of classes in the dynamic class descriptor table
- Change the list of security categories
- Change the list of security levels

To have changes take effect after altering a generic profile if the class is not RACLISTed using the RACROUTE REQUEST=LIST, GLOBAL=YES, or SETROPTS RACLISTER, one of the following steps is required:

- The security administrator issues the SETROPTS command:
  
  ```
  SETROPTS GENERIC(class-name) REFRESH
  ```

  See the SETROPTS command for authorization requirements.
- The user of the resource logs off and logs on again.

To have changes take effect after altering a generic profile if the class has been RACLISTed, the security administrator issues the following command:

  ```
  SETROPTS RACLISTER(class-name) REFRESH
  ```

**Attention:**

- When the RALTER command is issued from ISPF, the TSO command buffer (including SESSKEY, SSIGNON and possible BINDPW password data) is written to the ISPLOG data set. As a result, you should not issue this command from ISPF or you must control the ISPLOG data set carefully.
- When the command is issued as a RACF operator command, the command (including SESSKEY, SSIGNON and possible BINDPW password data) is written to the system log. Therefore, if any of the sensitive operands are used the command should be issued through TSO, not as an operator command.

**Issuing options**

The following table identifies the eligible options for issuing the RALTER command:

<table>
<thead>
<tr>
<th>As a RACF TSO command?</th>
<th>As a RACF operator command?</th>
<th>With command direction?</th>
<th>With automatic command direction?</th>
<th>From the RACF parameter library?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

For information on issuing this command as a RACF TSO command, refer to Chapter 3, “RACF TSO commands,” on page 15.

For information on issuing this command as a RACF operator command, refer to Chapter 4, “RACF operator commands,” on page 21.
RALTER

You must be logged on to the console to issue this command as a RACF operator command.

Related commands

- To define a general resource profile, see **RDEFINE (Define general resource profile)**.
- To list a general resource profile, see **RLIST (List general resource profile)**.
- To permit or deny access to a general resource profile, see **PERMIT (Maintain resource access lists)**.
- To obtain a list of general resource profiles, see **SEARCH (Search RACF database)**.

Authorization required

When issuing the RALTER command as a RACF operator command, you might require sufficient authority to the proper resource in the OPERCMDS class. See **z/OS Security Server RACF Security Administrator's Guide** for further information.

To alter the profile for a resource belonging to a class defined in the class descriptor table, you must have sufficient authority over the resource. RACF makes the following checks until one of the conditions is met:

- You have the SPECIAL attribute.
- The resource profile is within the scope of a group in which you have the group-SPECIAL attribute.
- You are the owner of the profile.
- If the profile is in the FILE or DIRECTRY class, the second qualifier of the profile name is your user ID.
- To assign a security label, you must have the SPECIAL attribute or have READ access to the security label profile. However, the security administrator can limit the ability to assign security labels to only users with the SPECIAL attribute.
- To assign a security category to a profile, you must have the SPECIAL attribute, or the access category must be in your user profile.
- To assign a security level to a profile, you must have the SPECIAL attribute, or, in your own profile, a security level that is equal to or greater than the security level you are assigning.
- Only a SPECIAL user can define a delegated resource (by specifying the RACF-DELEGATED string in the APPLDATA of the profile protecting the resource) when the resource has a SECLABEL and SETROPTS SECLABELCONTROL is in effect.
- To modify information in segments other than the base segment, such as DLFDATA, you must have the SPECIAL attribute or your installation must permit you to do so through field-level access checking.
- For a discrete profile, you are on the access list for the resource and you have ALTER authority. If you have any other level of authority, you cannot use the command for this resource.
- For a discrete profile, your current connect group (or, if list-of-groups checking is active, any group to which you are connected) is in the access list and has ALTER authority.
- For a discrete profile, the universal access authority for the resource is ALTER.

To use the GLOBALAUDIT operand, you must have the AUDITOR attribute or the profile is within the scope of a group in which you have group-AUDITOR attribute.
If you have the AUDITOR attribute or the resource profile is within the scope of a group in which you have the group-AUDITOR attribute, but you do not satisfy one of the above checks, you can specify only the GLOBALAUDIT operand.

Restrictions: The following operands have restrictions noted with the description of each operand:
- ADDMEM
- DELMEM
- ADDVOL
- GLOBALAUDIT.

To specify the AT keyword, you must have READ authority to the DIRECT.node resource in the RRSFDATA class and a user ID association must be established between the specified node.userid pair(s).

To specify the ONLYAT keyword you must have the SPECIAL attribute, the userid specified on the ONLYAT keyword must have the SPECIAL attribute, and a user ID association must be established between the specified node.userid pair(s) if the user IDs are not identical.

Syntax

For the key to the symbols used in the command syntax diagrams, see "Syntax of RACF commands and operands" on page 9. The complete syntax of the RALTER command is:

```
[subsystem-prefix][RALTER | RALT]

class-name
   (profile-name ...)
   [ ADDCATEGORY(category-name ...)
      | DELCATEGORY [ ([category-name ... | *) ] ]
      [ {ADDMEM | DELMEM} (member ...)
        | {ADDVOL | DELVOL} (volume-serial ...)
        | APPLDATA ('application-data') | NOAPPLDATA
        | AT([node].userid ...) | ONLYAT([node].userid ...)
        | AUDIT( access-attempt [audit-access-level] ... )
    ]
```
RALTER

[ CDTINFO(
    [ CASE ( UPPER | ASIS ) | NOCASE ]
    [ DEFAULTRC ( 0 | 4 | 8 ) | NODEFAULTRC ]
    [ DEFAULTUACC ( ACEE | ALTER | CONTROL
      | UPDATE | READ | NONE )
      | NODEFAULTUACC ]
    [ FIRST ( characters-allowed ... ) | NOFIRST ]
    [ GENERIC ( ALLOWED | DISALLOWED ) | NOGENERIC ]
    [ GENLIST ( ALLOWED | DISALLOWED ) | NOGENLIST ]
    [ GROUP ( grouping-class-name ) | NOGROUP ]
    [ KEYQUALIFIERS ( nnn ) | NOKEYQUALIFIERS ]
    [ MACPROCESSING ( NORMAL | REVERSE | EQUAL )
      | NOMACPROCESSING ]
    [ MAXLENGTH ( nnn ) | NOMAXLENGTH ]
    [ MAXLENX ( nnn ) | NOMAXLENX ]
    [ MEMBER ( member-class-name ) | NOMEMBER ]
    [ OPERATIONS ( YES | NO ) | NOOPERATIONS ]
    [ OTHER ( characters-allowed ... ) | NOOTHER ]
    [ POSIT ( nnn ) | NOPOSIT ]
    [ PROFILESALLOWED ( YES | NO )
      | NOPROFILESALLOWED ]
    [ RACLIST ( ALLOWED | DISALLOWED | REQUIRED )
      | NORACLIST ]
    [ SECLABELSREQUIRED ( YES | NO )
      | NOSECLABELSREQUIRED ]
    [ SIGNAL ( YES | NO ) | NOSIGNAL ]
  )
  | NOCDTINFO ]
[ CFDEF(
    [ FIRST( ALPHA | ALPHANUM | ANY
      | NONATABC | NONATNUM | NUMERIC ) ]
    [ HELP( help-text ) ]
    [ LISTHEAD( list-heading-text ) ]
    [ MAXLENGTH( maximum-field-length ) ]
    [ MAXVALUE( maximum-numeric-value ) | NOMAXVALUE ]
    [ MINVALUE( minimum-numeric-value ) | NOMINVALUE ]
    [ MIXED( YES | NO ) ]
    [ OTHER( ALPHA | ALPHANUM | ANY
      | NONATABC | NONATNUM | NUMERIC ) ]
  )
  | NOCFDEF ]
[ DATA ('installation-defined-data') | NODATA ]
[ DLFDATA(
    [ RETAIN® ( YES | NO ) | NORETAIN ]
    [ JOBNAMES(jobname1 ...)
      | NOJOBNAMES
    ]
    [ ADDJOBNAMES(jobname1 ...) ]
    [ DELJOBNAMES(jobname1 ...) ]
  )
  | NODLFDATA ]
[ EIM(
    [ DOMAINDN ( eim_domain_dn ) | NODOMAINDN ]
    [ OPTIONS ( ENABLE | DISABLE ) | NOOPTIONS ]
    [ LOCALREGISTRY ( registry_name ) | NOLOCALREGISTRY ]
    [ KERBRREGISTRY ( registry_name ) | NOKERBREGISTRY ]
    [ X509REGISTRY ( registry_name ) | NOX509REGISTRY ]
  )
  | NOEIM ]
[ GLOBALAUDIT ( access-attempt([audit-access-level]) ... ) ]
[ ICSF(
  [ ASYMUSAGE(
    [ HANDSHAKE | NOHANDSHAKE ]
    [ SECUREEXPORT | NOSECUREEXPORT ]
  )
  | NOASYMUSAGE ]
  [ SYMEXPORTABLE(BYANY | BYLIST | BYNONE)
    [ SYMEXPORTABLE ]
    [ SYMEXPORTEXPORTABLE ]
    [ SYMEXPORTCERTS([qualifier]/label-name ... | *)
      [ ADDSYMEXPORTCERTS([qualifier]/label-name ... | *)
      [ DELSYMEXPORTCERTS([qualifier]/label-name ... | *)
      [ NOSYMEXPORTCERTS ]
      [ SYMEXPORTKEYS(ICSF-key-label ... | *)
        [ ADDSYMEXPORTKEYS(ICSF-key-label ... | *)
        [ DELSYMEXPORTKEYS(ICSF-key-label ... | *)
        [ NOSYMEXPORTKEYS ]
    )
  )
  | NOICSF ]
[ ICTX(
  [ USEMAP ( YES | NO ) | NOUSEMAP ]
  [ DOMAP ( YES | NO ) | NODOMAP ]
  [ MAPREQUIRED ( YES | NO ) | NAMAPREQUIRED ]
  [ MAPPINGTIMEOUT (nnnn) | NOMAPPINGTIMEOUT ]
) )
[ NOICTX ]
[ KERB(
  [ DEFTKTLFE(def-ticket-life) | NODEFTKTLFE ]
  [ ENCRYPT(
    [ DES | NODES ]
    [ DES3 | NODES3 ]
    [ DESD | NODESD ]
    [ AES128 | NOAES128 ]
    [ AES256 | NOAES256 ]
  )
  | NOENCRYPT ]
  [ KERBNAME( kerberos-realm-name ) | NOKERBNAME ]
  [ MAXTTLFE(max-ticket-life) | NOMAXTTLFE ]
  [ MINTTLFE(min-ticket-life) | NOMINTTLFE ]
  [ PASSWORD( kerberos-password ) | NOPASSWORD ]
) )
[ NOKERB ]
[ LEVEL (nn) ]
[ NOTIFY [(userid)] | NONOTIFY ]
[ OWNER (userid or group-name) ]
[ PROXY(
  [ LDAPHOST (ldap_url) | NOLDAPHOST ]
  [ BINDDN (bind_distinguished_name) | NOBINDDN ]
  [ BINDPW (bind_password) | NOBINDPW ]
) ]
[ NOPROXY ]
[ SECLABEL (seclabel-name) | NOSECLABEL ]
[ SECLEVEL (seclevel-name) | NOSECLEVEL ]
RALTER

[ SESSION(
  [ CONVSEC( NONE | CONV | ALREADYV | PERSISTV | AVPV )
  | NOCONVSEC ]
  [ INTERVAL(n) | NOINTERVAL ]
  [ LOCK | NOLOCK ]
  [ SESSKEY(session-key) | NOSESSKEY ]
)
| NOSESSION ]

[ SIGVER(
  [ SIGREQUIRED( YES | NO ) | NOSIGREQUIRED ]
  [ FAILLOAD( ANYBAD | BADSIGNONLY | NEVER ) | NOFAILLOAD ]
  [ SIGAUDIT( ALL | SUCCESS | ANYBAD | BADSIGNONLY | NONE )
  | NOSIGAUDIT ]
  )
  ]

| NOSIGVER ]

[ SIGNEDSN | NOSIGNEDSN ]

[ SSIGNON(
  [ KEYMASKED(key-value)
  | KEYENCRYPTED(key-value) ]
)

| NOSSIGNON ]

[ STDATA(
  [ USER(userid | =MEMBER) | NOUSER ]
  [ GROUP(group-name | =MEMBER) | NOGROUP ]
  [ PRIVILEGED( YES | NO ) | NOPRIVILEGED ]
  [ TRACE( YES | NO ) | NOTRACE ]
  [ TRUSTED( YES | NO ) | NOTRUSTED ]
)

| NOSTDATA ]

[ SVFMR(
  [ SCRIPTNAME(script-name) | NOSCRIPTNAME ]
  | PARMNAME(parm-name) | NOPARMNAME ]
)

| NOSVFMR]

[ TIMEZONE( [E | W] hh [..mm] ) | NOTIMEZONE ]

[ TME(
  [ CHILDREN(profile-name ...)
  | ADDCHILDREN(profile-name ...)
  | DELCHILDREN(profile-name ...)
  | NOCHILDREN ]
  [ GROUPS(group-name ...)
  | ADDGROUPS(group-name ...)
  | DELGROUPS(group-name ...)
  | NOGROUPS ]
  [ PARENT(profile-name) | NOPARENT ]
  [ RESOURCE(resource-access-specification ...)
  | ADDRESOURCE(resource-access-specification ...)
  | DELRESOURCE(resource-access-specification ...)
  | NORESOURCE ]
  [ ROLES(role-access-specification ...)
  | ADDROLES(role-access-specification ...)
  | DELROLES(role-access-specification ...)
  | NOROLES ]
)

| NOTME ]

[ TVTOC | NOTVTOC ]

[ UACC(access authority) ]

[ WARNING | NOWARNING ]

[ WHEN( [ DAYS(day-info) ] [ TIME(time-info) ] ) ]
For information on issuing this command as a RACF TSO command, refer to Chapter 3, “RACF TSO commands,” on page 15.

For information on issuing this command as a RACF operator command, refer to Chapter 4, “RACF operator commands,” on page 21.

Parameters

**subsystem-prefix**

Specifies that the RACF subsystem is the processing environment of the command. The subsystem prefix can be either the installation-defined prefix for RACF (1–8 characters) or, if no prefix has been defined, the RACF subsystem name followed by a blank. If the command prefix was registered with CPF, you can use the MVS command D OPDATA to display it or you can contact your RACF security administrator.

Only specify the subsystem prefix when issuing this command as a RACF operator command. The subsystem prefix is required when issuing RACF operator commands.

**class-name**

Specifies the name of the class to which the resource belongs. Valid class names are those defined in the class descriptor table. For a list of general resource classes defined in the class descriptor table supplied by IBM, see Appendix B, “Supplied RACF resource classes,” on page 659.

This operand is required and must be the first operand following RALTER.

This command is not intended to be used for profiles in the following classes:

- DCEUUIDS
- DIGTCERT
- DIGTNMAP
- DIGTRING
- IDIDMAP
- NDSLINK
- NOTEILNK
- ROLE
- UNIXMAP.

**(profile-name ...)**

Specifies the name of the profile you want to change. The name you specify must be the name of an existing discrete or generic profile in the specified class. RACF uses the class descriptor table to determine the syntax of resource names within the class and whether the resource is a group.

Mixed-case profile names are accepted and preserved when `class-name` refers to a class defined in the static class descriptor table with CASE=ASIS or in the dynamic class descriptor table with CASE(ASIS).

This operand is required and must be the second operand following RALTER.

Notes:

1. For class TAPEVOL, if the volume serial specified for `profile-name` is a member of a tape volume set, then the profile definition for all tapes in the set is changed, since there is only one profile for the tape volume set.

   A tape volume set is used to refer to a set of two or more tapes created by the overflow of one tape to the next. RACF protects these tapes with one
profile. Hence, if the value specified for profile-name on this command is a member of a tape volume set, the changes in its resource profile affect the other members of the set.

2. You can specify only a single volume serial number if you also specify the ADDVOL or DELVOL operand.

3. To define a controlled program, you must specify class-name as PROGRAM and also specify ADDMEM or DELMEM. Also, you can specify only one profile-name.

4. If you specify class-name as PROGRAM, profile-name must identify one or more load modules or program objects. If you specify the full name of the program, the profile applies only to load modules or program objects with that specific name. If you specify the last character of the name as an *, the profile applies to all load modules or program objects that match the preceding part of the name, but only if they reside in one of the libraries listed in the profile's member list. For example, IKF* identifies all load module names that begin with IKF. If you specify profile-name as * or **, then the profile applies to all load modules and program objects that reside in one of the libraries you identify in the profile's member list, unless a profile with a more specific name and matching library applies.

5. For z/OS Integrated Security Services Network Authentication Service, the profile name for the definition of the local realm must be KERBDFLT.

6. RACF processes each profile name you specify independently, and all operands you specify apply to each named profile name. If an error occurs while processing a profile name, RACF issues a message and continues processing with the next profile name.

ADDCATEGORY | DELCATEGORY

ADDCATEGORY(category-name ...)

Specifies one or more names of installation-defined security categories. The category-name you specify must be defined as members of the CATEGORY profile in the SECDATA class. (For information on defining security categories, see z/OS Security Server RACF Security Administrator's Guide)

Specifying ADDCATEGORY causes RACF to add any category-name values you specify to any list of required categories that already exists in the resource profile. All users previously allowed to access the resource can continue to do so only if their profiles also include the additional values for category-name.

When the SECDATA class is active and you specify ADDCATEGORY, RACF performs security category checking in addition to its other authorization checking. If a user requests access to a resource, RACF compares the list of security categories in the user profile with the list of security categories in the resource profile. If RACF finds any security category in the resource profile that is not in the user's profile, RACF denies access to the resource. If the user's profile contains all the required security categories, RACF continues with other authorization checking.

Notes:

1. RACF does not perform security category checking for a started task with the RACF privileged or trusted attribute. The RACF privileged or trusted attribute can be assigned to a started task through the RACF started procedures table or STARTED class. Also, RACF does not enforce security category information specified on profiles in the PROGRAM class.
2. If you specify both ADDCATEGORY and DELCATEGORY, RACF uses the last operand that you specify.

DELCATEGORY[(category-name ... | *)]

Specifies one or more names of installation-defined security categories you want to delete from the resource profile. Specifying an asterisk (*) deletes all categories; RACF no longer performs security category checking for the resource.

Specifying DELCATEGORY by itself causes RACF to delete from the profile only undefined category names (those category names that were once known to RACF but that the installation has since deleted from the CATEGORY profile).

Note: If you specify both ADDCATEGORY and DELCATEGORY, RACF uses the last operand that you specify.

ADDMEM | DELMEM

Specifies the resource names that RACF is to add to, or delete from, the member list of the resource group profile indicated by profile-name.

ADDMEM(member ...)

You can use the ADDMEM operand to perform tasks such as altering security categories and security levels, entries in the global access checking table, and entries for program control, or to implement security labels on a system basis, as described in the following sections.

Mixed-case member names are accepted and preserved when class-name refers to a class defined in the static class descriptor table with CASE=ASIS or in the dynamic class descriptor table with CASE(ASIS). When class-name is GLOBAL and profile-name is the name of a class defined in the static class descriptor table with CASE=ASIS or in the dynamic class descriptor table with CASE(ASIS), the name part of a member entry in the GLOBAL access table is preserved as entered.

For ADDMEM with the DATASET class, no characters including generic characters, such as the asterisk (*) and the percent sign (%), can be combined with the value &RACUID to form a single qualifier level of the member name. This restriction does not exist for ADDMEM with classes other than DATASET.

For ADDMEM with the RACFVARS class, generic characters, such as the ampersand (&), the asterisk (*) and the percent sign (%) should not be specified in the member names.

To add members using the RALTER command, you need one of the following authorities, in addition to the authority needed to issue the RALTER command:

1. For classes other than SECLABEL, PROGRAM, SECDATA, GLOBAL, RACFVARS, and NODES, if the member resources are already RACF-protected by a member class profile or as a member of a profile in the same grouping class, one of the following must be true:
   • You have ALTER access authority to the member.
   • You are the owner of the member resource.
   • The member resource is within the scope of a group in which you have the group-SPECIAL attribute.
   • You have the SPECIAL attribute.
2. For classes other than SECLABEL, PROGRAM, SECDATA, GLOBAL, RACFVARS, and NODES, if the member resources are not RACF-protected (that is, there is no profile defined for that member), one of the following must be true:
   • You have CLAUTH authority to define resources in the member resource class.
   • You have the SPECIAL attribute.

3. To add a member to a profile in the RACFVARS or NODES class, one of the following must be true:
   • You have CLAUTH authority to define resources in the specified class (for example, RACFVARS or NODES).
   • You have the SPECIAL attribute.
   • You are the owner of the profile indicated by profile-name.
   • You have ALTER access authority to the profile indicated by profile-name.

4. To add a member to a profile in the SECLABEL, PROGRAM or SECDATA class, one of the following must be true:
   • You have CLAUTH authority to define resources in the specified class (for example, PROGRAM or SECDATA).
   • You have the SPECIAL attribute.

5. To add a member to a profile in the GLOBAL class (other than the GLOBAL DATASET, GLOBAL DIRECTRY, or GLOBAL FILE profile) using the following syntax:
   RALT GLOBAL class-name
   ADDMEM(resource-name/access-level)

   If the profile resource-name is already RACF-protected by a profile in class class-name:
   • You have ALTER access authority to the profile resource-name in class class-name.
   • You are the OWNER of the profile resource-name.
   • The profile resource-name in class class-name is within the scope of a group in which you have the group-special attribute.
   • You have the SPECIAL attribute.

6. If the profile resource-name is not already RACF-protected (that is, there is no profile defined for that member in class class-name):
   • You have CLAUTH authority to define resources in the class class-name.
   • You have the SPECIAL attribute.

7. To add a member to the GLOBAL DATASET profile, one of the following must be true:
   • You are the owner of the DATASET profile in the GLOBAL class.
   • The member is within the scope of a group in which you have the group-SPECIAL attribute, or the high-level qualifier of the member name is your user ID.
   • You have the SPECIAL attribute.

8. To add a member to the GLOBAL DIRECTRY or GLOBAL FILE profile, you must have the SPECIAL attribute.
For more information on the format of member names in general, and for specific classes (SECLABEL, GLOBAL, NODES, PROGRAM, SECDATA), see “Specifying member on the ADDMEM operand” on page 469.

**Note:** If you specify both ADDMEM and DELMEM, RACF uses the last operand that you specify.

**DELMEM**(member ...)  
Specifies the resource names that are to be deleted from the resource group indicated by profile-name. This operand is ignored if the class name specified is not a resource group class.

If class-name is specified as GLOBAL or PROGRAM, the rules for member are the same as given for ADDMEM. If class-name is specified as SECDATA, member should be a valid SECLABEL name or category name.

Mixed-case member names are accepted and preserved when class-name refers to a class defined in the static class descriptor table with CASE=ASIS or in the dynamic class descriptor table with CASE(ASIS). When class-name is GLOBAL and profile-name is the name of a class defined in the static class descriptor table with CASE=ASIS or in the dynamic class descriptor table with CASE(ASIS), the name part of a member entry in the GLOBAL access table is preserved as entered.

**Note:** If you specify both ADDMEM and DELMEM, RACF uses the last operand that you specify.

**ADDVOL | DELVOL**

**ADDVOL**(volume-serial ...)  
Specifies the tape volume serial numbers to be added to the tape volume set represented by profile-name. When you specify ADDVOL, profile-name must be a single volume serial number, which can identify any of the volumes currently defined in the volume set.

To use the ADDVOL operand, you must have the SPECIAL attribute, or you must have the CLAUTH attribute for the TAPEVOL resource class in addition to the other authorization requirements for using the RALTER command.

If you specify a generic profile, RACF ignores this operand.

**Notes:**

1. The ADDVOL operand is only valid for the TAPEVOL resource class.
2. If you specify both ADDVOL and DELVOL, RACF uses the last operand that you specify.

**DELVOL**(volume-serial ...)  
Specifies the tape volume serial numbers to be deleted from the tape volume set represented by profile-name. When you specify DELVOL, profile-name must be a single volume serial number, which can identify any of the volumes currently defined in the volume set except one of the volumes to be deleted. If you specify the same volume serial number for both profile-name and DELVOL, RACF ignores it.

If you try to delete a tape volume when the TAPEVOL profile contains one or more TVTOC entries, RACF does not complete the command if a TVTOC entry indicates that there is a protected data set on the volume. To delete this volume, you must first use the DELDSD command to delete any protected data sets on the volume.
RALTER

If you specify a generic profile, RACF ignores this operand.

Notes:
1. The DELVOL operand is only valid for the TAPEVOL resource class.
2. If you specify both ADDVOL and DELVOL, RACF uses the last operand that you specify.

APPLDATA | NOAPPLDATA

APPLDATA("application-data")
Specifies a text string that is associated with each of the named resources. The text string can contain a maximum of 255 characters and must be enclosed in single quotation marks. It can also contain double-byte character set (DBCS) data.

Rules:
• For profiles in the PROGRAM class, RACF examines the APPLDATA (if any) and perform special processing if you have specified MAIN or BASIC (optionally followed by blanks). This processing will occur only for profiles whose names do not end in *, and only when you have enabled enhanced PGMSECURITY mode. For details of this processing, see z/OS Security Server RACF Security Administrator’s Guide.
• For the FACILITY class, RACF examines the APPLDATA value of the following profiles:
  - BPX.UNIQUE.USER
    The APPLDATA value specifies the name of a user profile from which RACF can copy OMVS segment information (other than UID) when assigning unique UIDs through a callable service.
  - BPX.DEFAULT.USER
    The APPLDATA value specifies a user ID and group name from which RACF can retrieve default OMVS segment information. Beginning with z/OS Version 1 Release 11, the BPX.DEFAULT.USER profile is ignored when the BPX.UNIQUE.USER profile is defined.
  - BPX.NEXT.USER
    The APPLDATA value specifies information that RACF will use for the automatic assignment of OMVS UIDs and GIDs.
  - IRR.PGMSECURITY
    The APPLDATA value specifies whether RACF will operate in basic, enhanced, or enhanced-warning PGMSECURITY mode.
    - If the APPLDATA value contains the string ENHANCED, then RACF will run in enhanced PGMSECURITY mode.
    - If the APPLDATA value contains the string BASIC, then RACF will run in basic PGMSECURITY mode.
    - If the APPLDATA is empty or contains any other value, RACF will run in enhanced PGMSECURITY mode but in warning mode rather than failure mode.
  - IRR.PROGRAM.SIGNING.group.userid
  - IRR.PROGRAM.SIGNING.userid
  - IRR.PROGRAM.SIGNING.group
  - IRR.PROGRAM.SIGNING
    For any of the IRR.PROGRAM.SIGNING profiles, the APPLDATA value specifies the signing hash algorithm, and the SAF key ring to use when signing a program.
IRR.PROGRAM.SIGNATURE.VERIFICATION

The APPLDATA value specifies the SAF key ring to use when verifying the signature of a signed program.

- For the TIMS and GIMS class, specify application-data as REVERIFY to force the user to reenter his password whenever the transaction or transactions listed in the profile-name or ADDMEM operands are used.
- For the PTKTDATA class, the application-data field can be used to control the replay protection function of PassTicket support.
  - PassTicket replay protection prevents the use of user IDs to be shared among multiple users. However, in some events it is desirable to bypass this replay protection function.
  - Specifying no replay protection in the application-data field indicates that replay protection is to be bypassed. For example, the following command would result in replay protection being bypassed.

```
RALTER PTKTDATA profile-name
  APPLDATA('NO REPLAY PROTECTION')
```

Note the following:
- There must be a single space between the words no and replay, and between replay and protection. Lack of spaces or additional spaces or characters will make the command ineffective. For example, entering the following command would not result in replay protection being bypassed.

```
RALTER PTKTDATA profile-name
  APPLDATA('NOREPLAY PROTECTION')
```

- The text string no replay protection will always be translated to uppercase.
- The text string no replay protection can appear anywhere in the APPLDATA field.

- For the APPL class, when the APPLDATA value contains the RACF-INITSTATS(DAILY) string, RACF records statistics only for the first user verification of the day for the applications protected by this profile.
  The RACF-INITSTATS(DAILY) string is reserved text and may appear anywhere in the APPLDATA field. For more information about statistics collection, see [z/OS Security Server RACF Security Administrator's Guide](#)

- Specifying the RACF-DELEGATED string in the APPLDATA designates the resources protected by the profile as delegated, meaning that RACROUTE REQUEST=FASTAUTH should honor a nested ACEE during access checking to this resource. The RACF-DELEGATED string is reserved text and may appear anywhere in the APPLDATA field. For more information on nested ACEEs and delegated resources, see [z/OS Security Server RACF Security Administrator's Guide](#)

RACF does not validate the APPLDATA value during RALTER. Depending on the function, RACF might or might not issue any messages during subsequent processing if it finds an unexpected value.

The APPLDATA value, if present, can be displayed with the RLST command.

For detailed information about each APPLDATA value, see [z/OS Security Server RACF Security Administrator's Guide](#)

Chapter 5. RACF command syntax 413
NOAPPLDATA
Specifies that the RALTER command is to delete the text string that was
present in the profile associated with the resource.

AT | ONLYAT
The AT and ONLYAT keywords are only valid when the command is issued as a
RACF TSO command.

AT([node].userid ...)
Specifies that the command is to be directed to the node specified by node,
where it runs under the authority of the user specified by userid in the
RACF subsystem address space.

If node is not specified, the command is directed to the local node.

ONLYAT([node].userid ...)
Specifies that the command is to be directed only to the node specified by
node where it runs under the authority of the user specified by userid in the
RACF subsystem address space.

If node is not specified, the command is directed only to the local node.

AUDIT(access-attempt[(audit-access-level)])
Specifies which access attempts and access levels you want logged to the SMF
data set.

access-attempt
Specifies which access attempts you want logged to the SMF data set. The
following options are available:

ALL
   Specifies that you want to log both authorized accesses and detected
   unauthorized attempts to access the resource.

FAILURES
   Specifies that you want to log detected unauthorized attempts to access
   the resource.

NONE
   Specifies that you do not want any logging to be done for accesses to
   the resource.

SUCCESS
   Specifies that you want to log authorized accesses to the resource.

audit-access-level
Specifies which access levels you want logged to the SMF data set. The
levels you can specify are:

ALTER
   Logs ALTER access-level attempts only.

CONTROL
   Logs access attempts at the CONTROL and ALTER levels.

READ
   Logs access attempts at any level. This is the default value if no access
   level is specified.

UPDATE
   Logs access attempts at the UPDATE, CONTROL, and ALTER levels.

You cannot audit access attempts at the EXECUTE level.
CDTINFO | NOCDTINFO

CDTINFO
Specifies information used in the definition of an installation-defined class in the dynamic class descriptor table. CDTINFO should only be specified for profiles in the CDT class. Carefully plan changes to avoid unintended results. For guidelines, see “Guidelines for changing dynamic CDT entries” in z/OS Security Server RACF Security Administrator's Guide.

You can use the CDTINFO keyword with no operands to initiate validation checking of fields within the CDTINFO segment. For example, you issued an RDEFINE CDT command and received several errors, but you did not save a copy of the error messages. You could then issue the following command and the validation checking will be performed; those error messages will then be issued again.

RALTER CDT profile-name CDTINFO

CASE | NOCASE

CASE ( UPPER | ASIS )
Specifies whether mixed-case profile names are allowed for the class. When UPPER is specified, RACF translates the profile names for the specified class to uppercase. When ASIS is specified, RACF commands preserve the case of profile names for the specified class. Lowercase characters are allowed in any position of the profile name where alphabetic characters are allowed, based on the character restrictions in the FIRST and OTHER keywords.

NOCASE
Resets CASE to the default value of UPPER.

DEFAULTUACC | NODEFAULTUACC

DEFAULTUACC ( ALTER | CONTROL | UPDATE | READ | NONE )
Specifies the minimum access allowed if the access level is not set when a resource profile is defined in the class.

DEFAULTUACC ( ACEE )
If no universal access level is specified at the time the profile is created, RACF uses the default universal access authority from the command issuer's ACEE, as specified on the UACC operand of the ADDUSER, ALTUSER or CONNECT command.
NODEFAULTUACC
Resets the DEFAULTUACC to the default of NONE.

FIRST | NOFIRST
FIRST (characters-allowed ...)  
Specifies a character type restriction for the first character of the profile name. One or more of the following may be specified.

- **ALPHA**—Allows an alphabetic character (A–Z)
- **NUMERIC**—Allows a digit (0–9)
- **NATIONAL**—Allows characters # (X'7B'), @ (X'7C'), and $ (X'5B')
- **SPECIAL**—Allows any character except the following:
  - a blank
  - a comma
  - a parenthesis
  - a semicolon
  - those characters in ALPHA, NUMERIC, or NATIONAL.

NOFIRST
Resets FIRST to the default value of FIRST(ALPHA, NATIONAL).

GENERIC | NOGENERIC
GENERIC ( ALLOWED | DISALLOWED )
Specifies whether or not SETROPTS GENERIC and SETROPTS GENCMD are allowed for the class. The SETROPTS GENERIC command activates generic profile checking for a class. The SETROPTS GENCMD command activates generic profile command processing.

If GENERIC(DISALLOWED) is specified, GENLIST(ALLOWED) cannot also be specified.

Because generic processing is not allowed for grouping classes, GENERIC(DISALLOWED) should be specified if MEMBER(member-class-name) is also specified. If GENERIC(ALLOWED) is specified or defaulted for a grouping class, a warning message is issued. Subsequent processing for the dynamic class being defined and for profiles in that class will be treated as if GENERIC(DISALLOWED) was specified.

**Rule:** If the dynamic class you are defining shares a POSIT number with other classes, all classes with the shared POSIT number must have the same GENERIC keyword value. This is because the SETROPTS GENERIC and SETROPTS GENCMD commands process all classes that share a POSIT number. If at least one class specifies GENERIC(DISALLOWED) and at least one class specifies GENERIC(ALLOWED), RACF issues a warning message. When you subsequently add this class to the dynamic class descriptor table using the SETROPTS RACLST(CDT) command, RACF might change the value of the GENERIC keyword to match the GENERIC keyword value of the other classes sharing the POSIT number.

- If this dynamic class shares a POSIT number with an IBM-supplied class, RACF changes the value of the GENERIC keyword in the dynamic class to match the IBM class. (The class attribute in the IBM-supplied class takes precedence).
- If this dynamic class shares a POSIT number with an installation-defined class (static or dynamic), RACF determines
the least restrictive attribute—GENERIC(ALLOWED) is less restrictive than GENERIC(DISALLOWED)—and changes the GENERIC(DISALLOWED) class attribute to GENERIC(ALLOWED).

Exception: A grouping class and member class can share a POSIT number although their GENERIC keyword values need not match. You must specify GENERIC(DISALLOWED) for the grouping class. However, you can specify either ALLOWED or DISALLOWED for the member class.

NOGENERIC
Resets GENERIC to the default value of ALLOWED.

GENLIST | NOGENLIST

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

NOGENLIST
Resets GENLIST to the default value of DISALLOWED.

GROUP | NOGROUP

GROUP ( grouping-class-name )
Specifies the name of the class that groups the resources within the specified class. If GROUP is not specified, RACF does not allow resource grouping for the class. The grouping-class-name must be 1–8 characters.

When GROUP is specified, the class being defined is a member class.

If GROUP is specified, then grouping-class-name must also be defined in the CDT class, and its MEMBER keyword should refer to the class being defined. The GROUP and MEMBER keywords must have matching class entries before SETROPTS RACLST(CDT) is issued to build or refresh the dynamic CDT or before the system is restarted; otherwise, the class in error will not be added to the dynamic class descriptor table.

NOGROUP
Removes the grouping-class-name.

KEYQUALIFIERS | NOKEYQUALIFIERS

KEYQUALIFIERS ( nnn )
Specifies the number of matching qualifiers RACF uses when loading generic profile names to satisfy an authorization request if a discrete profile does not exist for a resource. For example, if you specify two for the class, all generic profile names whose highest level qualifiers match the two highest qualifiers of the entity name are loaded into the user's storage when the user requests access to a resource. The nnn value must be a number 0–123.
If KEYQUALIFIERS is not specified, the default is 0 and profile names for the entire class are loaded and searched.

The maximum value you can specify is 123, which is the maximum number of qualifiers in a name 246 characters long.

When KEYQUALIFIERS(nnn) is specified, generic profiles created in that class may not contain generic characters in the first nnn qualifiers of the profile name.

When KEYQUALIFIERS(nnn) is greater than 0 for a class, all discrete and generic profiles in that class must have at least nnn + 1 qualifiers in each profile name. The number of qualifiers a profile name is determined by counting the number of period characters in the profile and adding one; the first character is not examined.

Examples of valid profile names for KEYQUALIFIERS(2) are:

A.B.C
A.B.**
A.B.C.D*

**Guideline:** Specify KEYQUALIFIERS(nnn) greater than 0 for classes that have the following characteristics:

- The class is not usually RACLISTed or GENLISTed.
- Profile names in the class follow a naming convention where many generic profiles have the same nnn number of qualifiers at the beginning of the profile name.

For example, if you have an application that uses an installation-defined class to protect reports on terminal usage, you might have profiles such as these for each user on your z/OS system:

REPORTS.USER1.TERMUSE.*
REPORTS.USER1.TERMUSE.DEPT60.*
REPORTS.USER1.TERMUSE.2006.JAN.*
REPORTS.USER1.TERMUSE.2006.FEB.*
REPORTS.USER1.TERMUSE.2006.MAR.*
REPORTS.USER1.TERMUSE.2006.APR.*
REPORTS.USER1.TERMUSE.2006.MAY.*
REPORTS.USER1.TERMUSE.2006.JUN.*
REPORTS.USER1.TERMUSE.2006.JUL.*
REPORTS.USER1.TERMUSE.2006.AUG.*
REPORTS.USER1.TERMUSE.2006.SEP.*
REPORTS.USER1.TERMUSE.2006.OCT.*
REPORTS.USER1.TERMUSE.2006.NOV.*
REPORTS.USER1.TERMUSE.2006.DEC.*

In this example, you might define your installation class using KEYQUALIFIERS(3) so that when RACF checks authorization checks for resources in your class, only generic profile names that match the first three qualifiers of your report are loaded into storage for RACF to check.

**Restriction:** Different rules apply for the FILE and DIRECTRY classes. For the syntax required for profile names in the DIRECTRY and FILE classes, see the appropriate RACF Command Language Reference for your VM system.

**NOKEYQUALIFIERS**

Resets KEYQUALIFIERS to the default value of 0.
MACPROCESSING (NORMAL | REVERSE | EQUAL)
Specifies which type of mandatory access control (MAC) processing is required for the class:

- **NORMAL**—specifies normal MAC processing is required. If and when a MAC check is performed, the user's SECLABEL must dominate that of the resource.
- **REVERSE**—specifies reverse MAC processing is required. If and when a MAC check is performed, the SECLABEL of the resource must dominate that of the user.
- **EQUAL**—specifies equal MAC processing is required. If and when a MAC check is performed, the SECLABEL of the user must be equivalent to that of the resource.

MACPROCESSING(EQUAL) should be used for classes where two-way communication is expected. Writedown (SETROPTS MLS) does not apply to classes where MACPROCESSING(EQUAL) is specified.

NOMACPROCESSING
Resets MACPROCESSING to the default value of NORMAL.

MAXLENGTH | NOMAXLENGTH

MAXLENGTH (nnn)
Specifies the maximum length of resource and profile names for the specified class when MAXLENX is not specified. When MAXLENX is also specified, MAXLENGTH represents the maximum length of a resource name only when a RACROUTE macro is invoked with the ENTITY keyword. The value of **nnn** must be 1–246.

NOMAXLENGTH
Resets MAXLENGTH to the default value of 8.

MAXLENX | NOMAXLENX

MAXLENX (nnn)
Specifies the maximum length of resource and profile names for the specified class when a RACROUTE macro is invoked with the ENTITYX keyword or when a profile is added or changed using a RACF command processor. The value of **nnn** must be 1–246.

If MAXLENX is not specified before SETROPTS RACLIST(CDT) is issued to build or refresh the dynamic CDT or before the system is restarted, the value specified for MAXLENGTH is used for MAXLENX in subsequent processing for the dynamic class.

NOMAXLENX
Removes the MAXLENX value.

MEMBER | NOMEMBER

MEMBER (member-class-name)
Specifies the name of the class grouped by the resources within the specified class. The *member-class-name* value must be 1–8 characters.

When MEMBER is specified, the class being defined is a resource group.

If MEMBER is specified, then *member-class-name* must also be defined in the CDT class and its GROUP keyword should refer to the class being defined. The GROUP and MEMBER keywords must
have matching class entries before SETROPTS RACLIST(CDT) is issued to build or refresh the dynamic CDT or before the system is restarted; otherwise, the class in error will not be added to the dynamic class descriptor table.

NOMEMBER
Removes the member-class-name.

OPERATIONS | NOOPERATIONS

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

NOOPERATIONS
Resets OPERATIONS to the default value of NO.

OTHER | NOOTHER

OTHER (characters-allowed ...)
Specifies a character type restriction for the characters of the profile name other than the first character. One or more of the following may be specified:

- **ALPHA**—Allows an alphabetic character (A–Z)
- **NUMERIC**—Allows a digit (0–9)
- **NATIONAL**—Allows characters # (X'7B'), @ (X'7C'), and $ (X'5B')
- **SPECIAL**—Allows any character except the following:
  - a blank
  - a comma
  - a parenthesis
  - a semicolon
  - those characters in ALPHA, NUMERIC, or NATIONAL.

NOOTHER
Resets OTHER to the default of OTHER(ALPHA, NATIONAL).

POSIT | NOPOSIT

POSIT (nnn)
Specifies the POSIT number associated with the class. Each class in the class descriptor table has a POSIT number specified which identifies a set of option flags that control the following RACF processing options:

- Whether authorization checking should take place for the class (SETROPTS CLASSACT)
- Whether auditing should take place for resources within the class (SETROPTS AUDIT)
- Whether statistics should be kept for resources within the class (SETROPTS STATISTICS)
- Whether generic profile access checking is active for the class (SETROPTS GENERIC)
- Whether generic command processing is active for the class (SETROPTS GENCMD)
- Whether global access checking is active for the class (SETROPTS GLOBAL)
- Whether the user has CLAUTH to a resource class
Whether special resource access auditing applies to the class (SETROPTS LOGOPTIONS)
Whether SETROPTS RAclist will occur for this class (when RAclist(ALLOWED) or RAclist(REQUIRED) is also specified)
For all classes that have the same POSIT number specified, these options are identical. If you change an option for one class, this change will also affect all other classes that share the same POSIT number.

Before you issue SETROPTS RAclist(CDT) to build or refresh the dynamic class descriptor table, you must decide whether to use a unique set of option flags for each RACF class or whether to have two or more RACF classes share the same set of option flags. If you choose to use a unique set of option flags for a class, assign the class a unique POSIT number. If you choose to share the same set of option flags among several classes, assign those classes the same POSIT number.

Before you issue SETROPTS RAclist(CDT) to build or refresh the dynamic CDT, the POSIT keyword must specify a valid value on either the RDEFINE or RALTER command. Otherwise, the new or changed class will not be added to the dynamic class descriptor table.

Once you issue SETROPTS RAclist(CDT) to build or refresh the dynamic class descriptor table, you can activate the classes that comprise it and their respective set of option flags using the appropriate keywords on the SETROPTS command.

There are 1024 POSIT numbers that can identify 1024 sets of option flags. Installations can specify POSIT numbers 19–56 and 128–527. POSIT numbers 0–18, 57–127 and 528–1023 are reserved for IBM use and should not be specified for installation-defined classes unless an installation intends that one of its classes share SETROPTS options with an IBM-defined class.

Recommendation: A RACF class that has a default return code of 8 should not share a POSIT value with a RACF class having a different default return code. If a class with a default return code of 8 is activated but no profiles are defined, user activity that requires access in that class will be prevented.

NOPOSIT
Removes the POSIT number.

Before you issue SETROPTS RAclist(CDT) to build or refresh the dynamic CDT, the POSIT keyword must specify a valid value on either the RDEFINE or RALTER command. Otherwise, the new or changed class will not be added to the dynamic class descriptor table.

PROFILESALLOWED | NOPROFILESALLOWED

PROFILESALLOWED ( YES | NO )
Specifies whether you want RACF to allow profiles to be defined for this RACF class. If you specify PROFILESALLOWED(NO), RACF will not allow profiles to be defined to this RACF class; if a user attempts to define a profile to that class, the RDEFINE command responds with an appropriate message.
NOPROFILESALLOWED
Resets the PROFILESALLOWED value to the default of YES.

RACLIST | NORACLIST

RACLIST
Specifies whether SETROPTS RACLIST is to be allowed, disallowed or required for the specified class. If you process this class using SETROPTS RACLIST, RACF brings copies of all discrete and generic profiles within that class into storage in a data space. RACF uses those profiles in storage to check the authorization of any users who want to access the resources. The profiles remain in storage until removed by SETROPTS NORACLIST.

ALLOWED
Specifies that SETROPTS RACLIST may be used for the class, but is not required for authorization checking.

DISALLOWED
Specifies that SETROPTS RACLIST may not be used for the class.

REQUIRED
Specifies that you must process the class using SETROPTS RACLIST in order to use RACROUTE REQUEST=AUTH. The purpose of this keyword is to allow routines that cannot tolerate I/O to invoke RACF. When this keyword is specified and the class is not processed by SETROPTS RACLIST and a RACROUTE REQUEST=AUTH is attempted, the return code is 4.

NORACLIST
Resets the RACLIST value to the default of DISALLOWED.

SECLABELSREQUIRED | NOSECLABELSREQUIRED

SECLABELSREQUIRED ( YES | NO )
Specifies whether a SECLABEL is required for the profiles of the specified class when SETROPTS MLACTIVE is on.

SECLABELSREQUIRED(NO) means that RACF will not require a SECLABEL for profiles in this class; however, if a SECLABEL exists for this profile and the SECLABEL class is active, RACF will use it during authorization checking. SECLABELSREQUIRED(NO) applies to general resource classes that have no profiles, such as DIRAUTH, or for classes that contain no data, such as OPERCMDS and SECLABEL.

SECLABELSREQUIRED(YES) means that RACF will require a SECLABEL for profiles in this class when SETROPTS MLACTIVE is on.

NOSECLABELSREQUIRED
Resets the SECLABELSREQUIRED to the default of NO.

SIGNAL | NOSIGNAL

SIGNAL ( YES | NO )
Specifies whether an ENF signal should be sent to listeners when RACLISTed profiles are created, updated or deleted for authorization checking.
When SIGNAL(YES) is specified, RACF will send an ENF signal to
listeners when a SETROPTS RACLST, SETROPTS NORACLST
or SETROPTS RACLST REFRESH is issued for the class to
activate, deactivate, or update the profiles used for authorization
checking. For more information, see the [SIGNAL] section of [z/OS
Security Server RACF System Programmer's Guide]

When SIGNAL(NO) is specified, no ENF signal is sent.

SIGNAL(YES) is not valid if RACLST(DISALLOWED) is specified.

NOSIGNAL
Resets the SIGNAL value to the default of NO.

NOCDTINFO
Deletes the CDTINFO segment.

CFDEF | NOCFDEF

CFDEF
Changes the attributes of a custom field for profiles in the CFIELD class.
The custom fields you define with the CFDEF operand can be used in the
CSDATA segment of user and group profiles. For more information about
custom fields, including the profile name format, see [Defining and using
custom fields] in [z/OS Security Server RACF Security Administrator's
Guide] Changes in the custom field are not effective until the system programmer
rebuids the dynamic parse table using the IRRDPI00 UPDATE command.
For information about using the IRRDPI00 command, see [z/OS Security
Server RACF System Programmer's Guide]

You can use the CFDEF keyword with no suboperands to initiate validation
checking of fields within the CFDEF segment. For example, you issued an
RDEFINE CFIELD command and received several errors, but you did not
save a copy of the error messages. You could then issue the following
command and the validation checking will be performed; those error
messages will then be issued again.
RALTER CFIELD profile-name CFDEF

Rules:
• Specify CFDEF only for profiles in the CFIELD class.
• You cannot change the data type of a custom field using the RALTER
command. (Changing the data type might render the field unusable if all
other attributes are not correctly set.)

If you want to change the data type for a custom field, delete the CFIELD
profile using the RDELETE command, and then define the custom field
with the proper data type using the RDEFINE command.

Important: Plan carefully before you change the attributes of a custom
field. Most attributes are either required or desirable based on data type.
Therefore, you can change but not remove certain attributes using the
RALTER command.

FIRST
Specifies a character restriction for the first character in the custom
field.

Rules:
• You can change but you cannot remove the FIRST value.
The valid options for the FIRST attribute apply as follows, based on TYPE value (data type).

<table>
<thead>
<tr>
<th>Valid options</th>
<th>CHAR</th>
<th>FLAG</th>
<th>HEX</th>
<th>NUM</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALPHA</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ALPHANUM</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ANY</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NONATABC</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NONATNUM</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NUMERIC</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td></td>
</tr>
</tbody>
</table>

For each option of the FIRST attribute, the characters allowed in the custom field are as follows:

<table>
<thead>
<tr>
<th>Valid options</th>
<th>Alphabetic characters (A–Z)</th>
<th>National characters</th>
<th>Numeric characters (0–9)</th>
<th>Any other character</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALPHA</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>ALPHANUM</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>ANY</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>NONATABC</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>NONATNUM</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>NUMERIC</td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**ALPHA**
 Allows alphabetic characters (A–Z) and national characters # (X’7B’), @ (X’7C’), and $ (X’5B’).

**ALPHANUM**
 Allows alphabetic characters (A–Z), numbers (0–9), and national characters # (X’7B’), @ (X’7C’), and $ (X’5B’).

**ANY**
 Allows alphabetic characters (A–Z), numbers (0–9), national characters # (X’7B’), @ (X’7C’), and $ (X’5B’), and any other character. When you specify both FIRST(ANY) and OTHER(ANY), also allows quoted strings.

**NONATABC**
 Allows alphabetic characters, and excludes numbers and national characters # (X’7B’), @ (X’7C’), and $ (X’5B’).

**NONATNUM**
 Allows alphabetic characters and numbers, but excludes national characters # (X’7B’), @ (X’7C’), and $ (X’5B’).

**NUMERIC**
 Allows numbers (0–9).
HELP( help-text )
Specifies the help text for this custom field. The help text is displayed when the user is in TSO PROMPT mode and presses the PF1 key or enters a question mark (?). Lowercase alphabetic characters in the help-text value are translated to upper case.

Rules:
• Length: 1–255 characters.
• If the help text contains parentheses, commas, blanks, or semicolons, enclose the entire text string in single quotation marks.
• If a single quotation mark is intended to be part of the help text, use two single quotation marks together for each single quotation mark within the string, and enclose the entire string in single quotation marks.

Example: To define help text for a customer's address and indicate that the field can be up to 100 characters, you might specify the following value:
HELP('CUSTOMER'S ADDRESS. SPECIFY UP TO 100 CHARACTERS')
• You can change but you cannot remove the HELP value.

LISTHEAD( list-heading-text )
Specifies the heading to display in the output for the LISTUSER or LISTGRP command whenever the CSDATA segment is listed. Lowercase alphabetic characters in the list-heading-text value are translated to upper case.

Rules:
• Length: 1–40 characters.
• If the heading text contains parentheses, commas, blanks, or semicolons, enclose the entire text string in single quotation marks.
• If a single quotation mark is intended to be part of the help text, use two single quotation marks together for each single quotation mark within the string, and enclose the entire string in single quotation marks.

Example:
LISTHEAD('CUSTOMER'S ADDRESS =')
• You can change but you cannot remove the LISTHEAD value.

Guidelines: If you specify a LISTHEAD value, avoid confusion for users who use the LISTUSER or LISTGRP command to list custom field values by following these guidelines:
• Ensure that each custom field has a unique heading.
• Append an equal sign (=) or other delimiter to your LISTHEAD value to indicate in the list output where the heading ends and the data begins.

MAXLENGTH( maximum-field-length )
Specifies the maximum length of the custom field.

Rules:
• You can change but you cannot remove the MAXLENGTH value.
• The valid values or value ranges shown in Table 81 on page 426 apply based on data type.
Table 81. Valid values or value range for the MAXLENGTH keyword, based on data type

<table>
<thead>
<tr>
<th>Data type</th>
<th>Valid value or range</th>
</tr>
</thead>
<tbody>
<tr>
<td>CHAR</td>
<td>1–1100</td>
</tr>
<tr>
<td>FLAG</td>
<td>3</td>
</tr>
<tr>
<td>HEX</td>
<td>1–512</td>
</tr>
<tr>
<td>NUM</td>
<td>1–10</td>
</tr>
</tbody>
</table>

MAXVALUE | NOMAXVALUE

MAXVALUE( maximum-numeric-value )
Specifies the maximum numeric value for a custom field with TYPE(NUM).

Rules:
- Valid range: 0–2 147 483 647
- Do not specify a MAXVALUE value for custom fields with CHAR, FLAG, or HEX data type.
- Do not specify a MAXVALUE value lower than the MINVALUE value.
- Do not specify a MAXVALUE value longer than the highest value based on MAXLENGTH value.

NOMAXVALUE
Removes the MAXVALUE value. If you specify NOMAXVALUE, the following information is displayed when you list the CFDEF segment using the RLIST command.

MINVALUE | NOMINVALUE

MINVALUE( minimum-numeric-value )
Specifies the minimum numeric value for a custom field with TYPE(NUM).

Rules:
- Valid range: 0–2 147 483 647
- Do not specify a MINVALUE value for fields with CHAR, FLAG, or HEX data type.
- Do not specify a MINVALUE value higher than the MAXVALUE value.
- Do not specify a MINVALUE value longer than the highest value based on MAXLENGTH value.

NOMINVALUE
Removes the MINVALUE value. If you specify NOMINVALUE, the following information is displayed when you list the CFDEF segment using the RLIST command.

MIXED
Specifies whether mixed-case alphabetic characters are allowed for a custom field with TYPE(CHAR).

YES Lowercase characters are allowed in any position of the custom field where alphabetic characters are allowed, based on the
character restrictions specified in the FIRST and OTHER keywords. RACF commands, such as ADDUSER, do not translate lowercase alphabetic characters in the field to upper case.

**Rule:** Do not specify MIXED(YES) for custom fields with FLAG, HEX, or NUM data type.

**NO** RACF commands translate lowercase alphabetic characters in the field to upper case.

**OTHER**

Specifies a character restriction for characters in the custom field other than the first character.

For each option of the OTHER attribute, the characters allowed in the custom field are as follows:

<table>
<thead>
<tr>
<th>Valid options</th>
<th>Characters allowed</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Alphabet characters</td>
</tr>
<tr>
<td></td>
<td>(A–Z)</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>ALPHA</td>
<td>✓</td>
</tr>
<tr>
<td>ALPHANUM</td>
<td>✓</td>
</tr>
<tr>
<td>ANY</td>
<td>✓</td>
</tr>
<tr>
<td>NONATAABC</td>
<td>✓</td>
</tr>
<tr>
<td>NONATNUM</td>
<td>✓</td>
</tr>
<tr>
<td>NUMERIC</td>
<td></td>
</tr>
</tbody>
</table>

**ALPHA**

Allows alphabetic characters (A–Z) and national characters # (X’7B’), @ (X’7C’), and $ (X’5B’).

**ALPHANUM**

Allows alphabetic characters (A–Z), numbers (0–9), and national characters # (X’7B’), @ (X’7C’), and $ (X’5B’).

**ANY**

Allows alphabetic characters (A–Z), numbers (0–9), national characters # (X’7B’), @ (X’7C’), and $ (X’5B’), and any other character. When you specify both FIRST(ANY) and OTHER(ANY), also allows quoted strings.

**NONATAABC**

Allows alphabetic characters, and excludes numbers and national characters # (X’7B’), @ (X’7C’), and $ (X’5B’).

**NONATNUM**

Allows alphabetic characters and numbers, but excludes national characters # (X’7B’), @ (X’7C’), and $ (X’5B’).

**NUMERIC**

Allows numbers (0–9).

**Rules:**

- You can change but you cannot remove the OTHER value.
RALTER

- The valid options for the OTHER attribute apply as follows, based on TYPE value (data type).

<table>
<thead>
<tr>
<th>Valid options</th>
<th>Data type based on TYPE attribute</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>CHAR</td>
</tr>
<tr>
<td>ALPHA</td>
<td>✔</td>
</tr>
<tr>
<td>ALPHANUM</td>
<td>✔</td>
</tr>
<tr>
<td>ANY</td>
<td>✔</td>
</tr>
<tr>
<td>NONATABC</td>
<td>✔</td>
</tr>
<tr>
<td>NONATNUM</td>
<td>✔</td>
</tr>
<tr>
<td>NUMERIC</td>
<td>✔</td>
</tr>
</tbody>
</table>

**NOCFDEF**
Deletes the CFDEF segment.

**Important**: Avoid issuing the NOCFDEF operand for profiles in the CFIELD class because it causes the custom fields defined in the CFDEF segment to be unusable.

If you want to change the TYPE attribute, or remove an attribute that you are unable to remove using the RALTER command, delete the CFIELD profile using the RDELETE command, and then define the custom field with the proper attributes using the RDEFINE command.

**DATA | NODATA**

**DATA(‘installation-defined-data’)**
Specifies up to 255 characters of installation-defined data to be stored in the profile for the resource. The data must be enclosed in single quotation marks. It can also contain double-byte character set (DBCS) data.

This information is listed by the RLIST command.

**NODATA**
Specifies that the RALTER command is to delete the installation-defined data in the resource profile.

**DLFDATA | NODLFDATA**

**DLFDATA**
For profiles in the DLFCLASS, specifies information used in the control of DLF objects.

**RETAI(NYES | NO) | NORETAI(N**
Specifies whether the DLF object can be retained after use.

**JOBNAMES | NOJOBNAMES | ADDJOBNAMES | DELJOBNAMES**
You can specify any job name valid on your system. You can also specify generic job names with an asterisk (*) as the last character of a job name, to indicate generic job names. For example, JOBNAMES(ABC) allows only job ABC to access the DLF objects protected by the profile. JOBNAMES(ABC*) allows any job whose name begins with ABC (such as ABC, ABC1, or ABCDEF and so forth) to access to the DLF objects.

**JOBNAMES(jobname1 ...)**
Specifies the list of job names that can access the DLF objects protected by this profile.
NOJOBNAMES
   Specifies that no job names can access the DLF objects protected by this profile.

ADDJOBNAMES(jobname1...)
   Adds to the list of job names, the job names that can access the DLF objects protected by this profile.

DELJOBNAMES(jobname1...)
   Deletes the names from the job names list.

NODLFDATA
   Deletes the DLFDATA in the specified segment.

EIM | NOEIM

EIM
   The EIM and PROXY segment keywords and subkeywords combine to define the EIM domain, the LDAP host it resides on, and the bind information required by the EIM services to establish a connection with an EIM domain. The EIM services will attempt to retrieve this information when it is not explicitly supplied with the invocation parameters.

DOMAINDN | NODOMAINDN

DOMAINDN(eim_domain_dn)
   Specifies the distinguished name of the EIM domain. A valid EIM domain distinguished name begins with ibm-eimDomainName=.
   Uppercase and lowercase characters are accepted and maintained in the case in which they are entered.

   The EIM domain distinguished name is one component of an EIM domain name. An EIM domain name identifies the LDAP server that stores the EIM domain information. The EIM domain name begins with the ldap_url from the LDAPHOST suboperand of the keyword, followed by / and ends with the eim_domain_dn from the DOMAINDN suboperand. The length of a valid EIM domain name is determined by the combination of those factors. RACF allows the input of 1023 characters for the domain distinguished name. RACF does not ensure that an EIM domain name created from the LDAP URL and EIM domain distinguished name forms a valid EIM domain name.

   For more information about LDAP distinguished names, see IBM Tivoli Directory Server Administration and Use for z/OS or z/OS Integrated Security Services LDAP Server Administration and Use.

NODOMAINDN
   Deletes the eim_domain_dn value.

OPTIONS | NOOPTIONS
   Specifies options that control the EIM configuration.

ENABLE | DISABLE

   ENABLE
      Specifies that new connections may be established with the specified EIM domain. This is the default.

   DISABLE
      Specifies that new connections may not be established with the specified EIM domain.
NOOPTIONS
Resets OPTIONS to the default value of ENABLE.

LOCALREGISTRY | NOLOCALREGISTRY

LOCALREGISTRY(\textit{registry\_name})

Specifies the name of the local RACF registry in EIM domains. This operand is valid only with the following profiles and is ignored for all others:

- The IRR.PROXY.DEFAULTS profile in the FACILITY class
- The IRR.ICTX.DEFAULTS.sysid profile in the LDAPBIND class
- The IRR.ICTX.DEFAULTS profile in the LDAPBIND class.

EIM uses the \textit{registry\_name} value defined in the IRR.PROXY.DEFAULTS profile. The ICTX identity cache uses the \textit{registry\_name} value defined in the IRR.ICTX.DEFAULTS.sysid or IRR.ICTX.DEFAULTS profile.

The \textit{registry\_name} value is 1–255 characters in length. It can consist of any characters and can be entered with or without single quotation marks. The following rules apply:

- If parentheses, commas, blanks, or semicolons are intended as part of the \textit{registry\_name}, you must enclose the entire character string in single quotation marks.

- If a single quotation mark is intended as part of the \textit{registry\_name}, use two single quotation marks together for each single quotation mark within the string, and enclose the entire string within single quotation marks.

- Both uppercase and lowercase characters are accepted and maintained in the case in which they are entered.

NOLOCALREGISTRY

Deletes the local registry name from the profile. It does not affect the in-storage copy of the registry name. IPL the system to remove the in-storage copy.

KERBREGISTRY | NOKERBREGISTRY

KERBREGISTRY(\textit{registry\_name})

Specifies the name of the Kerberos registry in the EIM domain that the system is configured to use. This operand is only valid for the IRR.PROXY.DEFAULTS FACILITY class profile. The value is ignored when used on other profiles.

The Kerberos \textit{registry\_name} may be 1–255 characters in length. The \textit{registry\_name} can consist of any characters and can be entered with or without single quotation marks. The following rules apply:

- If parentheses, commas, blanks, or semicolons are to be entered as part of the \textit{registry\_name}, the character string must be enclosed in single quotation marks.

- If a single quotation mark is intended to be part of the \textit{registry\_name}, use two single quotation marks together for each single quotation mark within the string, and enclose the entire string within single quotation marks.

- Both uppercase and lowercase characters are accepted and maintained in the case in which they are entered.
NOKERBREGISTRY
Deletes the Kerberos registry name from the profile.

X509REGISTRY | NOX509REGISTRY
X509REGISTRY(registry_name)
Specifies the name of the X.509 registry in the EIM domain that the
system is configured to use. This operand is only valid for the
IRR.PROXY.DEFAULTS FACILITY class profile. The value is
ignored when used on other profiles.

The X.509 registry_name may be 1–255 characters long. The
registry_name can consist of any characters and can be entered
with or without single quotation marks. The following rules apply:

- If parentheses, commas, blanks, or semicolons are to be entered
  as part of the registry_name, the character string must be
  enclosed in single quotation marks.
- If a single quotation mark is intended to be part of the
  registry_name, use two single quotation marks together for each
  single quotation mark within the string, and enclose the entire
  string within single quotation marks.
- Both uppercase and lowercase characters are accepted and
  maintained in the case in which they are entered.

NOX509REGISTRY
Deletes the X.509 registry name from the profile.

NOEIM
Deletes the EIM segment.

GLOBALAUDIT(access-attempt[|audit-access-level])
Specifies which access attempts and access levels the user who has the
AUDITOR attribute wants logged to the SMF data set.

access-attempt
Specifies which access attempts the user who has the AUDITOR attribute
wants to log on the SMF data set.

ALL
Specifies that you want to log both authorized accesses and detected
unauthorized attempts to access the resource.

FAILURES
Specifies that you want to log detected unauthorized attempts to access
the resource.

NONE
Specifies that you do not want any logging to be done for accesses to
the resource.

SUCCESS
Specifies that you want to log authorized accesses to the resource.

audit-access-level
Specifies which access levels the user who has the AUDITOR attribute
wants to log on the SMF data set.

ALTER
Logs ALTER access-level attempts only.

CONTROL
Logs access attempts at the CONTROL and ALTER levels.
READ
Logs access attempts at any level. This is the default value if no access level is specified.

UPDATE
Logs access attempts at the UPDATE, CONTROL, and ALTER levels.

You cannot audit access attempts at the EXECUTE level.

To use GLOBALAUDIT, you must have the AUDITOR attribute, or the resource profile must be within the scope of a group in which you have the group-AUDITOR attribute.

Regardless of the value you specify for GLOBALAUDIT, RACF always logs all access attempts specified on AUDIT.

ICSF | NOICSF

ICSF
Specifies ICSF attributes for the keys that are controlled by this profile.
ICSF attributes are valid only for profiles in the CSFKEYS, GCSFKEYS, XCSFKEY, and GXCSFKEY classes.

ASYMUSAGE | NOASYMUSAGE

ASYMUSAGE
Specifies how an asymmetric key that is controlled by this profile is eligible to be used. If you do not specify ASYMUSAGE, the key is eligible for all uses.

SECUREEXPORT | NOSECUREEXPORT
Specifies whether the key is eligible to be used to export or import symmetric keys.

HANDSHAKE | NOHANDSHAKE
Specifies whether the key is eligible to be used to protect communication channels.

NOASYMUSAGE
Resets this option to the default setting. The key is eligible for all uses.

SYMEXPORTABLE | NOSYMEXPORTABLE

SYMEXPORTABLE
Specifies which public keys, if any, are eligible to be used to export a symmetric key that is controlled by this profile. If you do not specify SYMEXPORTABLE, any public key is eligible.

BYANY
Any public key is eligible. The SYMEXPORTCERTS and SYMEXPORTKEYS settings are ignored. This option is the default setting.

BYLIST
Only public keys specified with the SYMEXPORTCERTS or SYMEXPORTKEYS option are eligible. If neither option is set for this symmetric key, no public key is eligible (as if BYNONE were specified).

BYNONE
No public key is eligible. The SYMEXPORTCERTS and SYMEXPORTKEYS settings are ignored.
NOSYMEXPORTABLE
Resets the SYMEXPORTABLE option to BYANY.

SYMEXPORTCERTS | NOSYMEXPORTCERTS

SYMEXPORTCERTS([qualifier]/label-name ... | *)
Specifies a list of the labels of digital certificates that are eligible to be used to export the symmetric keys controlled by this profile.

Each listed certificate must exist in the ICSF key store (the SAF key ring or PKCS #11 specified by an ICSF configuration setting). For information about the ICSF key store, see z/OS Cryptographic Services ICSF Administrator’s Guide.

Specify an asterisk (*) to indicate that any certificate in the ICSF key store is eligible to be used to export the symmetric keys controlled by this profile. Specifying an asterisk (*) overrides any listed labels.

Specify each certificate label using a certificate label string in the form of qualifier/label-name.

<table>
<thead>
<tr>
<th>qualifier</th>
<th>The qualifier value is ...</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAF key ring</td>
<td>The RACF user ID of the certificate owner.</td>
</tr>
<tr>
<td>PKCS #11 token</td>
<td>The value of the CKA_ID attribute of the certificate. The CKA_ID value consists of up to 64 hexadecimal characters. Valid characters are 0–9 and A–F.</td>
</tr>
</tbody>
</table>

/label-name
Specifies the certificate label assigned when the certificate was created. You must specify the forward slash character (/) followed by the certificate label.

If the certificate label contains blanks, or special characters that cause problems with TSO/E, such as the comma, parenthesis, or comment delimiter (/*), the entire certificate label string must be enclosed in single quotation marks.

Any leading or trailing blanks specified in label-name are removed from this value before storing it in the profile.

Examples of certificate label strings:
DENICE/CertForDenice
'ROGERS/Cert for Rogers'
'/DLR cert'

ADDSYMEXPORTCERTS([qualifier]/label-name ... | *)
Adds the specified certificate labels to the current list of labels.

DELSYMEXPORTCERTS([qualifier]/label-name ... | *)
Removes the specified certificate labels from the current list of labels.
NOSYMEXPORTCERTS
   Removes the entire list of certificate labels.

SYMEXPORTKEYS | NOSYMEXPORTKEYS

SYMEXPORTKEYS(ICSF-key-label ... | *)
   Specifies a list of the ICSF key labels of public keys that are eligible
to be used to export the symmetric keys controlled by this profile.
Each listed public key must reside in the ICSF PKA key data set
(PKDS).
   Specify an asterisk (*) to indicate that any public key in the ICSF
PKDS is eligible to be used to export the symmetric keys controlled
by this profile. Specifying an asterisk (*) overrides any listed labels.

   ICSF-key-label
       Specifies the ICSF key label for the public key. The label name
cannot exceed 64 characters. The first character must be an
alphabetical character or a national character (#, @, or $).
   Subsequent characters can be a period character (.) or any
alphanumeric or national character.

ADDSYMEXPORTKEYS(ICSF-key-label ... | *)
   Adds the specified key labels to the current list of labels.

DELSYMEXPORTKEYS(ICSF-key-label ... | *)
   Removes the specified key labels from the current list of labels.

NOSYMEXPORTKEYS
   Removes the entire list of key labels.

NOICSF
   Deletes the ICSF segment.

ICTX | NOICTX

ICTX
   Specifies the ICTX configuration options that control the ICTX identity
cache.

   The ICTX identity cache uses an in-storage copy of the configuration
options. Use SETROPTS RA CLIST processing for the LDAPBIND class to
activate these options. (See the z/OS Security Server RACF Security
Administrator’s Guide for more information about SETROPTS RA CLIST
processing.)

   For details about the ICTX configuration options, see z/OS Integrated
Security Services EIM Guide and Reference.

   The following operands are used only for the following profiles in the
LDAPBIND class and are ignored for other profiles:
   - IRR.ICTX.DEFA ULT.S.sysid
   - IRR.ICTX.DEFA U LTS

USEMAP (YES I NO)
   Specifies whether the ICTX identity cache stores an identity mapping to
a local z/OS user ID when provided by the application.

   YES
       When the application provides a valid mapping to a local z/OS user
ID, the ICTX identity cache stores it.
NO
Identity mappings provided by the application are not stored.

NOUSEMAP
Resets the USEMAP value to YES.

DOMAP (YES | NO)
Specifies whether the ICTX identity cache uses Enterprise Identity Mapping (EIM) services to find a mapping to a z/OS user ID for an authenticated user, and then stores the mapping.

YES
When EIM finds a mapping to a z/OS user ID for an authenticated user, the ICTX identity cache stores it.

NO
The ICTX identity cache does not use EIM to find an identity mapping.

NODOMAP
Resets the DOMAP value to NO.

MAPREQUIRED (YES | NO)
Specifies whether the ICTX identity cache requires identity mapping to a z/OS user ID for an authenticated user.

YES
The ICTX identity cache fails the request when no valid mapping is provided by the application or found using EIM.

NO
The ICTX identity cache does not fail the request when no valid mapping is provided by the application or found using EIM.

NOMAPREQUIRED
Resets the MAPREQUIRED value to NO.

MAPPINGTIMEOUT (1–3600)
Specifies how long (one second to one hour) the ICTX identity cache stores an identity mapping to a z/OS user ID for an authenticated user.

Guideline: If you frequently modify your EIM mappings, consider a low MAPPINGTIMEOUT value. A shorter timeout period causes the ICTX identity cache to invoke EIM more frequently. This allows your cached mappings to be refreshed more frequently and improves their currency.

NOMAPPINGTIMEOUT
Resets the MAPPINGTIMEOUT value to 3600 seconds (one hour).

NOICTX
Deletes the ICTX segment.

KERB | NOKERB

KERB
Specifies z/OS Integrated Security Services Network Authentication Service information for a REALM class profile.

DEFTKTLFE | NODEFTKTLFE

DEFTKTLFE (def-ticket-life)
Specifies the default ticket lifetime for the local z/OS Network Authentication Service in seconds. The value for DEFTKTLFE is 1–2 147 483 647. Note that 0 is not a valid value.
RALTER

This keyword is only applicable when defining the KERBDFLT REALM profile for the local realm.

The RALTER command only requires specification of all of the ticket lifetime keywords on the same command invocation if RALTER is being used to initially define these values. If values have been previously defined, RACF uses both the previous values and new values specified to verify the specified def-ticket-life value.

NODEFKTLFE
Deletes the def-ticket-lifetime value for the local z/OS Network Authentication Service.

ENCRIPT | NOENCRYPT

ENCRIPT
Specifies which keys can be used by the z/OS Network Authentication Service realm you are changing.

DES | NODES
Whether DES encrypted keys can be used.

DES3 | NODES3
Whether DES3 encrypted keys can be used.

DESD | NODESD
Whether DESD encrypted keys can be used.

AES128 | NOAES128
Whether AES128 encrypted keys can be used.

AES256 | NOAES256
Whether AES256 encrypted keys can be used.

When a realm's password changes, a key of each type is generated and stored in the principal's user profile. The use of each key is based on the z/OS Network Authentication Service configuration.

See z/OS Integrated Security Services Network Authentication Service Administration for information about how z/OS Network Authentication Service uses keys and how to customize environment variables related to keys.

NOENCRYPT
Specifies that there is no restriction on which generated keys the realm can use, and resets the KERB ENCRYPT values to the default settings.

See z/OS Integrated Security Services Network Authentication Service Administration for information about how z/OS Network Authentication Service uses keys and how to customize environment variables related to keys.

KERBNAME | NOKERBNAME

KERBNAME(kerberos-realm-name)
Specifies the local realm name or a trust relationship for z/OS Network Authentication Service. The maximum length of this field is 117 characters.

- When you specify the local realm name for the KERBDFLT realm, you must specify KERBNAME using the unqualified form of the local realm name. For example:

  RALTER REALM KERBDFLT KERB(KERBNAME(KRB2000.IBM.COM))
**Important:** Avoid renaming your local realm name. If you rename your local realm, the keys for existing principals become unusable.

- When you specify a trust relationship, you must specify the *fully qualified* principal name using the following form:
  
  ```
  /.../kerberos_realm_name_1/krbtgt/kerberos_realm_name_2
  ```

  For more information about defining trust relationships, see [z/OS Integrated Security Services Network Authentication Service Administration](https://www.ibm.com/docs/en/zos/2.5.0?topic=krbconf-defining-trust-relationships).

**Syntax rules for naming your local realm:**

The local realm name that you define to RACF can consist of any character, except the `/ (X'61') character. You can enter the name with or without single quotation marks, depending on the following:

- If parentheses, commas, blanks, or semicolons are entered as part of the name, the character string must be enclosed in single quotation marks.
- If a single quotation mark is intended to be part of the name and the entire character string is enclosed in single quotation marks, you must use two single quotation marks together to represent each single quotation mark within the string.
- If the first character of the name is a single quotation mark, you must enter the string within single quotation marks, with two single quotation marks entered for the single quotation mark.

Regardless of the case in which it is entered, RACF translates the name of the local z/OS Network Authentication Service realm to upper case. However, RACF does not ensure that a valid `kerberos-realm-name` has been specified.

**Guidelines for naming your local realm:**

- Avoid using EBCDIC variant characters to prevent problems with different code pages.
- Carefully consider the length of the local realm name. Its length limits the length of local principal names because fully qualified local principal names use the following form and cannot exceed 240 characters:
  
  ```
  /.../kerberos_realm_name/principal_name
  ```

  The length of the fully qualified local principal name is checked by RACF only when a local `kerberos-principal-name` is added or altered. Therefore, plan ahead to ensure that the maximum length of your principal names is sufficient and help you avoid renaming the local realm. If you rename your local realm (using the RALTER command), the keys for existing principals become unusable.

**NOKERBNNAME**

Deletes the `kerberos-realm-name` value.

**MAXTKTLFE**

- **NOMAXTKTLFE**
  
  **MAXTKTLFE(max-ticket-life)**

  Specifies the `max-ticket-life` for the local z/OS Integrated Security Services Network Authentication Service in seconds. The value for MAXTKTLFE is 1–2 147 483 647. Note that 0 is not a valid value.
RALTER

This keyword is only applicable when defining the KERBDFLT REALM profile for the local z/OS Network Authentication Service realm.

The RALTER command only requires specification of all of the ticket lifetime keywords on the same command invocation if RALTER is being used to initially define these values. If values have been previously defined, RACF uses both these previous values and new values specified on the command, to verify the specified max-ticket-life value.

NOMAXTKTLFE
Deletes the max-ticket-lifetime value for the local z/OS Network Authentication Service.

MINTKTLFE | NOMINTKTLFE

MINTKTLFE(min-ticket-life)
Specifies the min-ticket-life for the z/OS Network Authentication Service in seconds. The value of MINTKTLFE is 1–2 147 483 647. Note that 0 is not a valid value.

This keyword is only applicable when defining the KERBDFLT REALM profile for the local realm.

The RALTER command only requires specification of all of the ticket lifetime keywords on the same command invocation if RALTER is being used to initially define these values. If values have been previously defined, RACF uses both the previous values and new values specified on the command to verify the specified min-ticket-life value.

NOMINTKTLFE
Deletes the min-ticket-lifetime value for the local z/OS Network Authentication Service principal.

PASSWORD | NOPASSWORD

PASSWORD(kerberos-password)
Specifies the value of the kerberos-password. The maximum length of this value is 128 characters. The PASSWORD keyword is applicable to all REALM class profile definitions. A password must be associated with the definition of a trust relationship or else the definition is incomplete.

Guideline: Avoid using EBCDIC variant characters to prevent problems with different code pages.

The kerberos-password that you define to RACF might consist of any character. You can enter a password with or without single quotation marks, depending on the following:

- If parentheses, commas, blanks, or semicolons are entered as part of the password, the character string must be enclosed in single quotation marks.
- If a single quotation mark is intended to be part of the password and the entire character string is enclosed in single quotation marks, you must use two single quotation marks together for each single quotation mark within the string.
RALTER

- If the first character of the password is a single quotation mark, you must enter the string within single quotation marks, with two single quotation marks entered for the character.

Both uppercase and lowercase characters are accepted and maintained in the case in which they are entered.

**Note:** This keyword is intended for administrators to be able to associate a password with the definition of a realm. It is not the same as a RACF user password and is not constrained by the SETROPTS password rules and change interval values that might be established for RACF user passwords.

**NOPASSWORD**
Deletes the z/OS Network Authentication Service password. If this is the local z/OS Network Authentication Service realm (KERBDFLT), it will no longer be able to grant ticket-granting tickets. Removal of the password from a foreign realm definition will invalidate the inter-realm trust relationship.

**NOKERB**
Deletes the KERB segment.

**LEVEL(nn)**
Specifies a level indicator, where nn is an integer in the range of 00–99. Your installation assigns the meaning of the value. It is included on all records that log resource accesses and is listed by the RLIST command.

**NOTIFY | NONOTIFY**

**NOTIFY[(userid)]**
Specifies the user ID of a RACF-defined user to be notified whenever RACF uses this profile to deny access to a resource. If you specify NOTIFY without specifying a user ID, RACF takes your user ID as the default; you are notified whenever the profile denies access to a resource.

If you receive NOTIFY messages, you should log on frequently to take action in response to the unauthorized access attempt described in each message. RACF sends NOTIFY messages to the SYS1.BROADCAST data set. When the resource profile also includes WARNING, RACF might have granted access to the resource to the user identified in the message.

When RACF denies access to a resource, it does not notify a user:
- When the resource is in the PROGRAM class
- When the resource is in a class for which an application has built in-storage profiles using RACROUTE REQUEST=LIST

Some applications, such as IMS™ and CICS, load all the profiles for a given class into storage. After these profiles are in storage, the applications can do a fast authorization check using RACROUTE REQUEST=FASTAUTH. Fast authorization checking is different from normal authorization checking in several ways. One difference is that, in some cases, fast authorization checking does not issue warning messages, notification messages or support auditing. In cases where it does not, return and reason codes are returned to the application to allow support of these functions. The application can examine the return and reason codes and use RACROUTE REQUEST=AUTH to create the messages and audit records. If the application uses RACROUTE
REQUEST=AUTH to support auditing or specifies LOG=ASIS on RACROUTE REQUEST=FASTAUTH, the specified user is notified. Otherwise, notification, warning, and such do not occur.

For details on using RACF with IMS, see z/OS Security Server RACF Security Administrator's Guide. For details on using RACF with CICS, refer to CICS RACF Security Guide.

- When the profile is used to disallow the creation or deletion of a data set NOTIFY is used only for resource access checking, not for resource creation or deletion.

NONOTIFY
Specifies that no user is to be notified when RACF uses this profile to deny access to a resource.

OWNER(userid or group-name)
Specifies a RACF-defined user or group to be assigned as the new owner of the resource you are changing.

To change the owner of a resource, you must be the current owner of the resource or have the SPECIAL attribute, or the profile must be within the scope of a group in which you have the group-SPECIAL attribute. The user specified as the owner does not automatically have access to the resource. Use the PERMIT command to add the owner to the access list as desired.

PROXY | NOPROXY
PROXY
Specifies information which the z/OS LDAP server will use when acting as a proxy on behalf of a requester. The R_proxyserv (IRRSPY00) SAF callable service will attempt to retrieve this information when it is not explicitly supplied with the invocation parameters. Applications or other services which use the R_proxyserv callable service, such as IBM Policy Director Authorization Services for z/OS and OS/390, may instruct their invaders to define PROXY segment information.

LDAPHOST | NOLDAPHOST
LDAPHOST(ldap_url)
Specifies the URL of the LDAP server which the z/OS LDAP server will contact when acting as a proxy on behalf of a requester. An LDAP URL has a format such as ldap://12.34.56.78:389 or ldaps://12.34.56.78:636, where ldaps indicates that an SSL connection is desired for a higher level of security. LDAP will also allow you to specify the host name portion of the URL using either the text form (BIGHOST.POK.IBM.COM) or the dotted decimal address (12.34.56.78). The port number is appended to the host name, separated by a colon (X'7A').

For more information about LDAP URLs and how to enable LDAP servers for SSL connections, see IBM Tivoli Directory Server Administration and Use for z/OS or z/OS Integrated Security Services LDAP Server Administration and Use.

The LDAP URL that you define to RACF can consist of 10–1023 characters. A valid URL must start with either ldap:// or ldaps:///. RACF will allow any characters to be entered for the remaining portion of the URL, but you should ensure that the URL conforms to TCP/IP conventions. For example, parentheses, commas, blanks,
semicolons, and single quotation marks are not typically allowed in
a host name. The LDAP URL can be entered with or without single
quotation marks.

RACF does not ensure that a valid LDAP URL has been specified.

NOLDAPHOST
Deletes the URL of the LDAP server which the z/OS LDAP server
will contact when acting as a proxy on behalf of a requester.

BINDDN | NOBINDDN

BINDDN(bind_distinguished_name)
Specifies the distinguished name (DN) which the z/OS LDAP server
will use when acting as a proxy on behalf of a requester. This DN
will be used in conjunction with the BIND password, if the z/OS
LDAP server needs to supply an administrator or user identity to
BIND with another LDAP server. A DN is made up of attribute: value
pairs, separated by commas. For example:

cn=Ben Gray,ou=editing,o=New York Times,c=US
cn=Lucille White,ou=editing,o=New York Times,c=US
cn=Tom Brown,ou=reporting,o=New York Times,c=US

When you define a BIND DN to RACF, it can contain 1–1023
characters. The BIND DN can consist of any characters and can be
entered with or without single quotation marks. The following rules
apply:

• If parentheses, commas, blanks, or semicolons are to be entered
  as part of the BIND DN, the character string must be enclosed in
  single quotation marks.
• If a single quotation mark is intended to be part of the BIND DN,
  use two single quotation marks together for each single quotation
  mark within the string, and enclose the entire string within single
  quotation marks.

Both uppercase and lowercase characters are accepted and
maintained in the case in which they are entered. For more
information about LDAP distinguished names, see IBM Tivoli
Directory Server Administration and Use for z/OS or z/OS
Integrated Security Services LDAP Server Administration and Use

If you issue the RALTER command as a RACF operator command
and you specify the BIND DN in lowercase, you must include the
BIND DN within single quotations.

RACF does not ensure that a valid BIND DN has been specified.

NOBINDDN
Deletes the distinguished name (DN) used by the z/OS LDAP
server when acting as a proxy on behalf of a requester.

BINDPW | NOBINDPW

BINDPW
Specifies the password which the z/OS LDAP server will use when
acting as a proxy on behalf of a requester.

When you define a BIND password to RACF, it can contain 1–128
characters. The BIND password can consist of any characters (see
exception below) and can be entered with or without single
quotation marks. The following rules apply:
The BIND password cannot start with a left brace \( \{ \) character (X’8B’).

If parentheses, commas, blanks, or semicolons are to be entered as part of the BIND password, the character string must be enclosed in single quotation marks.

If a single quotation mark is intended to be part of the BIND password, use two single quotation marks together for each single quotation mark within the string, and enclose the entire string within single quotation marks.

Both uppercase and lowercase characters are accepted and maintained in the case in which they are entered. For more information about LDAP passwords, see IBM Tivoli Directory Server Administration and Use for z/OS or z/OS Integrated Security Services LDAP Server Administration and Use.

If you issue the RALTER command as a RACF operator command and you specify the BIND password in lowercase, you must include the BIND password within single quotations.

RACF does not ensure that a valid BIND password has been specified.

**NOBINDPW**

Deletes the password used by the z/OS LDAP server when acting as a proxy on behalf of a requester.

**NOPROXY**

Deletes LDAP proxy information.

**SECLABEL **\( \) NOSECLABEL

**SECLABEL(seclabel-name)**

Specifies an installation-defined security label for this profile. A security label corresponds to a particular security level (such as CONFIDENTIAL) with a set of zero or more security categories (such as PAYROLL or PERSONNEL).

If you are authorized to use the SECLABEL, RACF stores the name of the security label you specify in the resource profile.

If you are not authorized to the SECLABEL or if the name you had specified is not defined as a SECLABEL profile in the SECLABEL class, the resource profile is not updated. If the SECLABEL class is active and the security level is specified in this profile, any security levels and categories in the profile are ignored.

**NOSECLABEL**

Removes the security label, if one had been specified, from the profile.

**SECLEVEL **\( \) NOSECLEVEL

**SECLEVEL(seclevel-name)**

Specifies the name of an installation-defined security level. This name corresponds to the number that is the minimum security level that a user must have to access the resource. The `seclevel-name` must be a member of the SECLEVEL profile in the SECDATA class.

When you specify SECLEVEL and the SECDATA class is active, RACF adds security level access checking to its other authorization checking. If global access checking does not grant access, RACF compares the security level allowed in the user profile with the security level required in
the resource profile. If the security level in the user profile is less than the security level in the resource profile, RACF denies the access. If the security level in the user profile is equal to or greater than the security level in the resource profile, RACF continues with other authorization checking.

RACF does not perform security level checking for a started task that has the RACF privileged or trusted attribute. The RACF privileged or trusted attribute can be assigned to a started task through the RACF started procedures table or STARTED class. Also, RACF does not enforce security level information specified on profiles in the PROGRAM class.

If the SECDATA class is not active, RACF stores the name you specify in the resource profile. When the SECDATA class is activated and the name you specified is defined as a SECLEVEL profile, RACF can perform security level access checking for the resource profile. If the name you specify is not defined as a SECLEVEL profile, you are prompted to provide a valid seclevel-name.

NOSECLEVEL
Specifies that the RALTER command is to delete the security level name from the profile. RACF no longer performs security level checking for the resource.

SESSION | NOSESSION
SESSION
Controls the establishment of sessions between logical units under LU6.2. This operand is only valid for the APPCLU resource class. It allows the following suboperand to add, change, or delete SESSION segment field values when changing an APPCLU class profile.

CONVSEC | NOCONVSEC
CONVSEC(security-checking-level)
Specifies the level or levels of security checking performed when conversations are established with the LU protected by this profile.

The security-checking-level value can be one of the following levels.

Guideline: Specify a CONVSEC option for each APPCLU profile.

NONE
All inbound allocate requests pass without RACF checking for a valid user ID. No RACROUTE REQUEST=VERIFY is issued.

CONV
APPC/MVS issues a RACROUTE REQUEST=VERIFY to verify the user ID and password for all inbound allocate requests.

ALREADYV
APPC/MVS RACF does not verify the user ID and password for any inbound allocate requests. If you specify ALREADYV, you assume that user IDs and passwords have already been verified by the partner LU. You must specify this only if the partner LU is trustworthy.

PERSISTV
Specifies persistent verification.

AVPV
The user ID/password is already verified and persistent verification is requested. In general, you should select one of NONE, CONV, and ALREADYV for each APPCLU profile.
NOCONVSEC
Delete any existing conversation security parameters.

INTERVAL | NOINTERVAL
INTERVAL(n)
Sets the maximum number of days the session key is valid. This value of n is 1–32767. If the key interval is longer than the installation maximum (set with SETROPTS SESSIONINTERVAL), the INTERVAL is not changed.

NOINTERVAL
There is no limit on the number of days the key is valid.

LOCK | NOLOCK
LOCK
Marks the profile as locked.

NOLOCK
Unlocks a previously locked profile.

SESSKEY | NOSESSKEY
SESSKEY(session-key)
Changes the key for this profile. The session-key value can be expressed in two ways:
• X’y’ where y is a hexadecimal number of 1–16 digits
• z or ‘z’ where z is a string of 1–8 characters
If the entire 16 digits or 8 characters are not used, the field is padded to the right with binary zeros.

Note: Session keys are 64-bit Data Encryption Standard (DES) keys. With DES, 8 of the 64 bits are reserved for use as parity bits, so those 8 bits are not part of the 56-bit key. In hexadecimal notation, the DES parity bits are: X’0101 0101 0101 0101’. Any two 64-bit keys are equivalent DES keys if their only difference is in one or more of these parity bits. For instance, the following SESSKEY values, although appearing to be quite different, are equivalent because they differ only in the last bit of each byte:
• BDF0KM4Q, which is X’C2C4 C6F0 D2D4 F4D8’
• CEG1LN5R, which is X’C3C5 C7F1 D3D5 F5D9’

NOSESSKEY
Deletes the session key for this profile.

NOSESSION
Deletes the SESSION segment from this profile.

SIGVER | NOSIGVER
SIGVER
Specifies the options for verifying the signatures of programs that are protected by this general resource profile.

Rule: Specify SIGVER only for profiles in the PROGRAM class. Any options specified with the SIGVER operand are ignored for profiles in a class other than the PROGRAM class.
Restriction: Digital signature verification is supported only for program objects that are stored as members of a partitioned data set extended (PDSE) library. Digital signature verification is not supported for programs that are stored as members of a partitioned data set (PDS) library.

Any options specified with the SIGVER operand are ignored for unsupported programs.

Note: Regardless of the SIGREQUIRED setting, specifying FAILLOAD(NEVER) and SIGAUDIT(NONE) is equivalent to having no SIGVER segment.

For detailed information, see “Program signing and verification” in Z/OS Security Server RACF Security Administrator’s Guide.

SIGREQUIRED | NOSIGREQUIRED

SIGREQUIRED

Specifies whether programs that are protected by this profile must be digitally signed.

YES

Specifies that programs must be digitally signed.

When you specify SIGREQUIRED(YES), the following conditions apply to any program that is protected by this general resource profile:

- If the program has a digital signature:
  - Signature verification processing occurs.
  - The program continues to load according to the FAILLOAD setting.
  - Logging occurs according to the SIGAUDIT setting.
- If the program has no digital signature:
  - Signature verification processing occurs, resulting in a signature verification failure.
  - The program continues to load according to the FAILLOAD setting.
  - Logging occurs according to the SIGAUDIT setting.

NO

Specifies that programs need not be digitally signed.

When you specify SIGREQUIRED(NO), the following conditions apply to any program that is protected by this general resource profile:

- If the program has a digital signature:
  - Signature verification processing occurs.
  - The program continues to load according to the FAILLOAD setting.
  - Logging occurs according to the SIGAUDIT options.
- If the program has no digital signature:
  - No signature verification occurs.
  - The program continues to load. The FAILLOAD setting is ignored.
  - No logging occurs. The SIGAUDIT setting is ignored.
**FAILLOAD | NOFAILLOAD**

FAILLOAD

Specifies the conditions under which the program fails to load in the event that a signature verification failure occurs.

ANYBAD

Specifies that the program fails to load when a signature verification failure occurs, regardless of the cause. Such failures include those resulting from an incorrect signature, or an error establishing the trust of the signer. This setting includes failures related to administrative errors, such as a missing or incorrectly defined key ring.

The ANYBAD setting includes the failures covered by the BADSIGONLY setting, and also includes errors establishing the trust of the signer.

BADSIGONLY

Specifies that the program fails to load only when the signature verification failure is caused by an incorrect digital signature. Such failures include only those resulting from a signature that fails verification or a signature structure that is missing or improperly formatted.

In contrast to ANYBAD, the BADSIGONLY setting does not cause a program to fail to load when the program has a valid signature originating from an untrusted signer.

NEVER

Specifies that the program never fails to load when a signature verification failure is detected.

NOFAILLOAD

Resets the FAILLOAD value to NEVER.

**SIGAUDIT | NOSIGAUDIT**

SIGAUDIT

Specifies which signature verification events are logged. Messages are issued to the console only for signature verification failures that are logged.

ALL

Logs all signature verifications, whether successful or not.

SUCCESS

Logs only signature verification successes. In other words, the digital signature is valid and the root CA certificate is trusted.

ANYBAD

Logs all signature verification failures, regardless of the cause of the failure. Such failures include those resulting from an incorrect signature, or an error establishing the trust of the signer. This setting includes failures related to administrative errors, such as a missing or incorrectly defined key ring.

The ANYBAD setting logs the failures covered by the BADSIGONLY setting, and also logs errors encountered when establishing the trust of the signer.
BADSIGONLY
Logs only signature verification failures caused by an incorrect digital signature. Such failures include only those resulting from a signature that fails verification or a signature structure that is missing or improperly formatted.

In contrast to ANYBAD, the BADSIGONLY setting does not log a signature verification failure when the program has a valid signature originating from an untrusted signer.

NONE
Logs no digital signature verification events.

NOSIGAUDIT
Resets the SIGAUDIT value to NONE.

NOSIGVER
Deletes the SIGVER segment.

SINGLEDSN | NOSINGLEDSN
SINGLEDSN
Specifies that the tape volume can contain only one data set. SINGLEDSN is valid only for a TAPEVOL profile. If the volume already contains more than one data set, RACF issues a message and ignores the operand.

NOSINGLEDSN
Specifies that the tape volume can contain multiple data sets, up to a maximum of 9999. NOSINGLEDSN is valid only for a TAPEVOL profile.

SSIGNON | NOSSIGNON
SSIGNON
Defines the application key or a secured signon key and indicates the method you want to use to protect the key value within the RACF database. You can mask or encrypt the key. The key-value represents a 64-bit (8-byte) key that must be represented as 16 hexadecimal characters. The valid characters are 0–9 and A–F.

Notes:
1. As with RACF passwords, the database unload facility does not unload application keys or a secured signon keys.
2. The RLIST command does not list the value of the application key or the secured signon keys. Therefore, when you define the keys, you should note the value and keep it in a secure place.

KEYMASKED(key-value)
Specifies that you want to mask the key value using the masking algorithm.

Notes:
1. You can specify this operand only once for each application key.
2. If you mask a key, you cannot encrypt it. These are mutually exclusive.

You can use the RLIST command to verify that the key is protected.

KEYENCRYPTED(key-value)
Specifies that you want to encrypt the key value.

Notes:
1. You can specify this operand only once for each application key.
2. If you encrypt a key, you cannot mask it. These are mutually exclusive.
3. A cryptographic product must be installed and active on the system.

You can use the RLIST command to verify that the key is protected.

NOSSIGNON
Specifies that the SSIGNON segment should be deleted.

STDATA | NOSTDATA
STDATA
Used to control security for started tasks. STDATA should only be specified for profiles in the STARTED class.

USER | NOUSER

USER(userid)
Specifies the user ID to be associated with this entry.

RACF issues a warning message if the specified userid does not exist, but information is added to the STDATA segment. If the error is not corrected, RACF uses the started procedures table to process START requests that would have used this STARTED profile.

USER(=MEMBER)
Specifies that the procedure name should be used as the user ID. If =MEMBER is specified for USER, a group-name value should be specified for the GROUP operand. If =MEMBER is specified for both USER and GROUP, a warning message is issued and problems might result when the profile is used. For information, see z/OS Security Server RACF Security Administrator's Guide.

NOUSER
Specifies the user ID should be deleted from this entry, leaving it unspecified. A warning message is issued because the absence of a user specification in the STDATA segment normally indicates that the segment information is incomplete. IF NOUSER is specified, RACF uses the started procedures table to process START requests that would have used this STARTED profile.

GROUP | NOGROUP

GROUP(group-name)
Specifies the group name to be associated with this entry.

RACF issues a warning message if the specified group-name does not exist. If userid and group-name are specified, RACF verifies that the user is connected to the group. If there is an error in the specification of the group name, the started task runs as an undefined user.

GROUP(=MEMBER)
Specifies that the procedure name should be used as the group name. If =MEMBER is specified for GROUP, a userid value must be specified for the USER operand or RACF uses the started procedures table to assign an identifier for this started task. If =MEMBER is specified for both USER and GROUP, a warning message is issued and problems might result when the profile is used. For information, see z/OS Security Server RACF Security Administrator's Guide.
NOGROUP
Specifies the group name should be deleted from this entry, leaving it unspecified. IF NOGROUP is specified, the started task runs with the default group of the specified user ID.

PRIVILEGED(YES | NO) | NOPRIVILEGED
Specifies whether the started task should run with the RACF PRIVILEGED attribute. The PRIVILEGED attribute allows the started task to pass most authorization checking. No installation exits are called, no SMF records are generated, and no statistics are updated. (Note that bypassing authorization checking includes bypassing the checks for security classification of users and data.) For more information, see z/OS Security Server RACF Macros and Interfaces.

PRIVILEGED(NO) and NOPRIVILEGED indicate that the started task should run without the PRIVILEGED attribute.

If neither PRIVILEGED nor NOPRIVILEGED is specified, PRIVILEGED(NO) is the default.

TRACE(YES | NO) | NOTRACE
Specifies whether a message should be issued to the operator when this entry is used to assign an ID to the started task.

If TRACE(YES) is specified, RACF issues an informational message to the operator to record the use of this entry when it is used to assign an ID to a started task. This record can be useful in finding started tasks that do not have a specific entry defined and in diagnosing problems with the user IDs assigned for started tasks.

TRACE(NO) and NOTRACE specify that an informational message should not be issued when this entry is used to assign an ID to the started task.

If neither TRACE nor NOTRACE is specified, TRACE(NO) is the default.

TRUSTED(YES | NO) | NOTRUSTED
Specifies whether the started task should run with the RACF TRUSTED attribute. The TRUSTED attribute is similar to the PRIVILEGED attribute except that auditing can be requested using the SETROPTS LOGOPTIONS command. For more information about the TRUSTED attribute, see z/OS Security Server RACF System Programmer's Guide.

TRUSTED(NO) and NOTRUSTED indicate that the started task should run without the RACF TRUSTED attribute.

If neither TRUST nor NOTRUSTED is specified, TRUSTED(NO) is the default.

NOSTDATA
Specifies that all the STDATA information for this entry should be deleted. When this entry is used, and no STDATA was specified (or when the STDATA has been deleted), then RACF issues a message and use the started procedures table to assign information for this START command.

SVFMR | NOSVFMR

SVFMR
Defines profiles associated with a particular SystemView for MVS application.

SCRIPTNAME | NOScriptname
**RALTER**

**SCRIPTNAME**(script-name)
Specifies the name of the list of default logon scripts associated with this application. This operand is optional. If you omit this operand, no scripts are changed for the application.

The *script-name* is the 1–8 character alphanumeric name of a member of an MVS partitioned data set (PDS). RACF accepts both uppercase and lowercase characters for *script-name*, but lowercase characters are translated to uppercase.

The PDS member specified by *script-name* contains a list of other PDS members that contain the scripts associated with this application’s profile. The PDS and members, including the member that contains the list of other members, are created by the administrator of the SystemView for MVS application.

**NOSCMNIGNAME**
Specifies that the logon script name should be deleted from this entry.

**PARMNAME | NOPARMNAME**

**PARMNAME**(parm-name)
Specifies the name of the parameter list associated with this application. This operand is optional. If this operand is omitted, no parameters are changed for the application.

The *parm-name* is the 1–8 character alphanumeric name of a member of an MVS partitioned data set (PDS). RACF accepts both uppercase and lowercase characters for *parm-name*, but lowercase characters are translated to uppercase.

The PDS member specified by *parm-name* contains a list of other PDS members that contain the parameters associated with this application’s profile. The PDS and members, including the list of other members, are created by System View for the MVS administrator.

**NOPARMNAME**
Specifies that the parameter list name should be deleted from this entry.

**NOSVFMR**
Specifies that the SVFMR segment should be deleted.

**TIMEZONE | NOTIMEZONE**

**TIMEZONE**( {E | W} hh[,mm])
Specifies the time zone in which a terminal resides. TIMEZONE is valid only for resources in the TERMINAL class; RACF ignores it for all other resources.

Specify TIMEZONE only when the terminal is not in the same time zone as the processor on which RACF is running. In this situation, TIMEZONE provides the information RACF needs to calculate the time and day values correctly. If you identify more than one terminal in the *profile-name* operand, all the terminals must be in the same time zone.

On TIMEZONE, you specify whether the terminal is east (E) or west (W) of the system and by how many hours (*hh*) and, optionally, minutes (*mm*). The terminal time zone is different from the processor time zone. Valid hour values are 0–11, and valid minute values are 00–59.
For example, if the processor is in New York and the terminal is in Los Angeles, specify \texttt{TIMEZONE(W 3)}. If the processor is in Houston and the terminal is in New York, specify \texttt{TIMEZONE(E 1)}.

If you change the local time on the processor (to accommodate daylight savings time, for instance), RACF adjusts its time calculations accordingly. However, if the processor time zone and the terminal time zone do not change in the same way, you must adjust the terminal time zones yourself, as described for the \texttt{WHEN(TIME)} operand.

\textbf{NOTIMEZONE}

Specifies that the terminal is in the same time zone as the processor. NOTIMEZONE is valid only for resources in the terminal class; RACF ignores it for all other resources.

\textbf{TME | NOTME}

\textbf{TME}

Specifies that information for the Tivoli Security Management Application is to be added, changed, or deleted.

\textbf{Note}: The TME segment fields are intended to be updated only by the Tivoli Security Management Application, which manages updates, permissions, and cross references. A security administrator should only directly update Tivoli Security Management fields on an exception basis.

All TME suboperands, with the exception of those for ROLES, can be specified when changing a resource profile in the ROLE class. Conversely, only the ROLES suboperand can be specified when changing a resource profile in any other class.

\textbf{CHILDREN | NOCHILDREN | ADDCHILDREN | DELCHILDREN}

\textbf{CHILDREN} (\textit{profile-name} ...)

Specifies the complete list of roles which inherit attributes from this role. A role is a discrete general resource profile defined in the ROLE class.

\textbf{ADDCHILDREN} (\textit{profile-name} ...)

Specifies the addition of specific child roles to the current list of roles.

\textbf{DELCHILDREN} (\textit{profile-name} ...)

Specifies the removal of specific child roles from the current list of roles.

\textbf{NOCHILDREN}

Specifies the removal of the entire list of child roles.

\textbf{GROUPS | NOGROUPS | ADDGROUPS | DELGROUPS}

\textbf{GROUPS} (\textit{group-name} ...)

Specifies the complete list of groups which should be permitted to resources defined in this role profile.

The \textit{group-name} value should be the name of a defined group.

\textbf{ADDP|NOTME

\textbf{TME | NOTME}

\textbf{TME}

Specifies that information for the Tivoli Security Management Application is to be added, changed, or deleted.

\textbf{Note}: The TME segment fields are intended to be updated only by the Tivoli Security Management Application, which manages updates, permissions, and cross references. A security administrator should only directly update Tivoli Security Management fields on an exception basis.

All TME suboperands, with the exception of those for ROLES, can be specified when changing a resource profile in the ROLE class. Conversely, only the ROLES suboperand can be specified when changing a resource profile in any other class.

\textbf{CHILDREN | NOCHILDREN | ADDCHILDREN | DELCHILDREN}

\textbf{CHILDREN} (\textit{profile-name} ...)

Specifies the complete list of roles which inherit attributes from this role. A role is a discrete general resource profile defined in the ROLE class.

\textbf{ADDCHILDREN} (\textit{profile-name} ...)

Specifies the addition of specific child roles to the current list of roles.

\textbf{DELCHILDREN} (\textit{profile-name} ...)

Specifies the removal of specific child roles from the current list of roles.

\textbf{NOCHILDREN}

Specifies the removal of the entire list of child roles.

\textbf{GROUPS | NOGROUPS | ADDGROUPS | DELGROUPS}

\textbf{GROUPS} (\textit{group-name} ...)

Specifies the complete list of groups which should be permitted to resources defined in this role profile.

The \textit{group-name} value should be the name of a defined group.

\textbf{ADDGROUPS} (\textit{group-name} ...)

Specifies the addition of specific groups to the current list of groups.
DELGROUPS(
  group-name ...)
  Specifies the removal of specific groups from the current list of groups.

NOGROUPS
  Specifies the removal of the entire list of groups.

PARENT | NOPARENT

  PARENT(profile-name)
  Specifies the name of a role from which this role inherits attributes.
  A role is a discrete general resource profile defined in the ROLE class.

  NOPARENT
  Specifies the deletion of the parent role from this profile.

RESOURCE | NORESOURCE | ADDRESOURCE | DELRESOURCE

  RESOURCE(resource-access-specification ...)
  Specifies the complete list of resources and associated access levels for groups defined in this role profile.

  Delresource(resource-access-specification ...)
  Specifies the removal of specific resource-access-specifications from the current list.

  NORESOURCE
  Specifies the removal of the entire list of resources.

ROLES | NOROLES | ADDROLES | DELROLES

  ROLES(role-access-specification ...)
  Specifies a list of roles and associated access levels related to this profile.

  One or more role-access-specification values can be specified, each separated by blanks. Each value should contain no imbedded blanks and should have the following format:
role-name:authority
    [:conditional-class:conditional-profile]

where role-name is a discrete general resource profile defined in the ROLE class. The authority value is the access authority (NONE, EXECUTE, READ, UPDATE, CONTROL, or ALTER) with which groups in the role definition should be permitted to the resource.

The conditional-class value is a class name (APPCPORT, CONSOLE, JESINPUT, PROGRAM, TERMINAL, or SYSID) for conditional access permission, and is followed by the conditional-profile value, a resource profile defined in the conditional class.

ADDOALES(role-access-specification ...)
    Specifies that specific roles and access levels are to be added to the current list.

DELROLES(role-access-specification ...)
    Specifies that specific roles from the current list of roles are to be removed.

NOROLES
    Specifies that the entire list of roles be removed.

NOTME
    Specifies that RACF delete the TME segment from the profile.

TVTOC | NOTVTOC

TVTOC
    Specifies, for a TAPEVOL profile, that RACF is to create a TVTOC in the TAPEVOL profile when a user creates the first output data set on the volume.

    Specifying TVTOC affects the access list for the TAPEVOL profile:
    1. When RACF processes the RALTER command with the TVTOC operand, it places the user ID of the command issuer (perhaps the tape librarian) in the access list with ALTER authority.
    2. When the first output data set is created on the volume, RACF adds the user ID associated with the job or task to the access list with ALTER authority.


    The TVTOC operand is valid only for a discrete profile in the TAPEVOL class. If you specify TVTOC and the volume already contains a TVTOC, RACF issues a message and ignores the operand.

NOTVTOC
    Specifies that RACF cannot create a TVTOC in the resource profile. The NOTVTOC operand is valid only for a discrete profile in the TAPEVOL class. It is also invalid if a TVTOC with at least one entry already exists in the TAPEVOL profile. When NOTVTOC is invalid, RACF issues a message and ignores the operand. If your installation uses DFSMShsm™ and you activate tape data set protection, the TVTOC for DFSMShsm tapes might become too large. To avoid this problem, issue the following RALTER command:

    RALTER TAPEVOL HSMHSM NOTVTOC
**UACC(access-authority)**

Specifies the universal access authority to be associated with this resource. The universal access authorities are ALTER, CONTROL, UPDATE, READ, EXECUTE (for controlled programs only), and NONE.

**Notes:**
1. For tape volumes and DASD volumes, RACF treats CONTROL authority as UPDATE authority.
2. For all other resources listed in the class descriptor table, RACF treats CONTROL and UPDATE authority as READ authority.
3. If a user accessing a data set has the RESTRICTED attribute, RACF treats the universal access authority (UACC) as NONE for that access attempt.

**WARNING | NOWARNING**

**WARNING**

Specifies that even if access authority is insufficient, RACF is to issue a warning message and allow access to the resource. RACF also records the access attempt in the SMF record if logging is specified in the profile.

**Restriction:** RACF does not issue a warning message for a resource when the resource is:
- In the PROGRAM or NODES class
- In a class for which an application has built in-storage profiles using RACROUTE REQUEST=LIST.

**When SETROPTS MLACTIVE(FAILURES) is in effect:** A user or task can access a resource that is in WARNING mode and has no security label even when MLACTIVE(FAILURES) is in effect and the class requires security labels. The user or task receives a warning message and gains access.

**Applications that use REQUEST=LIST:** Some applications, such as IMS and CICS, load all the profiles for a given class into storage. After these profiles are in storage, the applications can do a fast authorization check using RACROUTE REQUEST=FASTAUTH. Fast authorization checking is different from normal authorization checking in several ways. One difference is that, in some cases, fast authorization checking does not issue warning messages, notification messages or support auditing. In cases where it does not, return and reason codes are returned to the application to allow support of these functions. The application can examine the return and reason codes and use RACROUTE REQUEST=AUTH to create the messages and audit records. If the application uses RACROUTE REQUEST=AUTH to support auditing or specifies LOG=ASIS on the RACROUTE REQUEST=FASTAUTH, the specified user is notified. Otherwise, notification, warning, and so on, does not occur.

For details on using RACF with IMS, see z/OS Security Server RACF Security Administrator's Guide. For details on using RACF with CICS, refer to CICS RACF Security Guide.

**NOWARNING**

Specifies that if access authority is insufficient, RACF is to deny the user access to the resource and not issue a warning message.

**WHEN**

Specifies, for resources in the TERMINAL class, the days of the week or the hours in the day when the terminal can be used to access the system. The
day-of-week and time restrictions apply only when a user logs on to the system; that is, RACF does not force the user off the system if the end-time occurs while the user is logged on.

If you specify the WHEN operand, you can restrict the use of the terminal to certain days of the week or to a certain time period on each day. You can also restrict access to both certain days of the week and to a certain time period within each day.

**DAYS(day-info)**

Specifies days of the week when the terminal can be used. The *day-info* value can be any one of the following:

- **ANYDAY**
  Allows use of the terminal on any day.

- **WEEKDAYS**
  Allows use of the terminal only on weekdays (Monday through Friday).

  *day ...
  Allows use of the terminal only on the days specified, where *day* can be MONDAY, TUESDAY, WEDNESDAY, THURSDAY, FRIDAY, SATURDAY, or SUNDAY, and you can specify the days in any order.

**TIME(time-info)**

Specifies the time period each day when the terminal can be used. The *time-info* value can be any one of the following:

- **ANYTIME**
  RACF allows use of the terminal at any time.

  *start-time:* *end-time*
  RACF allows use of the terminal only during the specified time period. The format of both the *start-time* and *end-time* values is *hhmm*, where *hh* is the hour in 24-hour notation (00–24) and *mm* is the minutes (00–59) within the range 0001–2400. Note that 2400 indicates 12:00 a.m. (midnight).

  If *start-time* is greater than *end-time*, the interval spans midnight and extends into the following day.

  Specifying *start-time* and *end-time* is straightforward when the processor on which RACF is running and the terminal are in the same time zone; you specify the time values in local time.

  However, if the terminal is in a different time zone from the processor and you want to restrict access to certain time periods, you have two choices. You can specify the TIMEZONE operand to allow RACF to calculate the time and day values correctly. Or, you can adjust the time values yourself, by translating the *start-time* and *end-time* for the terminal to the equivalent local time for the processor.

  For example, assume that the processor is in New York and the terminal is in Los Angeles, and you want to allow access to the terminal from 8:00 A.M. to 5:00 P.M. in Los Angeles. In this situation, you would specify TIME(1100:2000). If the processor is in Houston and the terminal is in New York, you would specify TIME(0900:1800).

If you omit DAYS and specify TIME, the time restriction applies to any day-of-week restriction already specified in the profile. If you omit TIME and specify DAYS, the days restriction applies to any time restriction already
RALTER

specified in the profile. If you specify both DAYS and TIME, RACF allows use of the terminal only during the specified time period and only on the specified days.

Examples

Table 82. RALTER examples

| Example 1 | Operation | User TRA02 wants to change the owner and universal access for terminal TERMID01 and restrict use of the terminal to weekdays during regular business hours (8:00 A.M.–6:00 P.M.). |
| Known | User TRA02 has the SPECIAL attribute. |
| Command | RALTER TERMINAL TERMID01 OWNER(TRA02) UACC(ALTER) WHEN(DAYS(WEEKDAYS) TIME(0800:1800)) |
| Defaults | None. |

Example 2

| Operation | User RFF23 wants to delete the two data fields associated with the terminal T3E8. The user wants to be notified whenever the terminal profile denies access to the terminal. |
| Known | User RFF23, who is a RACF-defined user, is the owner of the T3E8 terminal entry. |
| Command | @RALTER TERMINAL T3E8 NODATA NOAPPLDATA NOTIFY(RFF23) |
| Defaults | None. |

Example 3

| Operation | User ADM1 wants to delete the data fields associated with the generic profile * in the TERMINAL class. |
| Known | User ADM1 has the SPECIAL attribute. |
| Command | RALTER TERMINAL * NODATA NOAPPLDATA |
| Defaults | None. |

Example 4

| Operation | User PAYADM1 wants to add the PAYROLL category to the list of security categories known to RACF. |
| Known | User PAYADM1 has the SPECIAL attribute. RACF security category checking is active. |
| Command | RALTER SECDATA CATEGORY ADDMEM(PAYROLL) |
| Defaults | None. |

Example 5

| Operation | User RFF22 wants to add volume TAP02 to the tape volume set, change the level of the tape volume set, and change the AUDIT and GLOBALAUDIT logging options. |
| Known | User RFF22 is the owner of the tape volume set. User RFF22 has the AUDITOR attribute. TAP01 is a volume of the tape volume set. User RFF22 wants to issue the command as a RACF TSO command. |
| Command | RALTER TAPEVOL TAP01 AUDIT(SUCCESS(READ)) LEVEL(22) GLOBALAUDIT(SUCCESS(UPDATE) FAILURES(READ)) ADDVOL(TAP02) |
| Defaults | None. |
Table 82. RALTER examples (continued)

Example 6  
**Operation**  
User ADM1 wants to add AMASPZAP to the in-storage profile table of controlled programs. AMASPZAP requires program-accessed data set checking.

**Known**  
User ADM1 has the SPECIAL attribute. AMASPZAP resides in SYS1.LINKLIB on the SYSRES volume. RACF program control is active.

User ADM1 wants to issue the command as a RACF TSO command.

**Command**  
RALTER PROGRAM AMASPZAP ADDMEM('SYS1.LINKLIB'/SYSRES/PADCHK)

**Defaults**  
None.

Example 7  
**Operation**  
User ADM1 wants to add all load modules that start with IKF to the in-storage profile table of controlled programs. These load modules do not require program-accessed data set checking. User ADM1 wants to direct the command to run at the local node under the authority of user EMILIE and prohibit the command from being automatically directed to other nodes.

**Known**  
Users ADM1 and EMILY have the SPECIAL attribute. All load modules whose names begin with IKF reside in SYS1.COBBLIB on the SYSRES volume. RACF program control is active. Users ADM1 and EMILIE have an already established user ID association.

User ADM1 wants to issue the command as a RACF TSO command.

**Command**  
RALTER PROGRAM IKF* ONLYAT(.EMILIE)  
ADDMEM('SYS1.COBBLIB'/SYSRES/NOPADCHK)

**Results**  
The command is only processed on the local node and not automatically directed to any other nodes in the RRSF configuration.

Example 8  
**Operation**  
The security administrator wants to change the key value of a profile in the PTKTDATA class so the value becomes encrypted.

**Known**  
NONNEL is the user ID of the security administrator. The profile name is TSOR004. The key-value is B004194019641980. The security administrator wants to issue the command as a RACF TSO command.

**Command**  
RALTER PTKTDATA TSOR004  
SSIGNON(KEYENCRYPTED(B004194019641980))

**Defaults**  
None.

Example 9  
**Operation**  
The administrator wants to change the script and parameter definitions for an existing SystemView for MVS application that has been defined to the SYSMVIEW class.

**Known**  
The new script definition is APPL2SC.

The new parameter definition is APPL2P.

**Command**  
RALTER SYSMVIEW APPL1.HOST1.USER1  
SVFMR(SCRIPTNAME(APPL2SC) PARMNAME(APPL2P))

**Defaults**  
None.

Example 10  
**Operation**  
Local realm KRB2000.IBM.COM is being defined with a minimum ticket lifetime of 5 minutes, a default ticket lifetime of 10 hours, a maximum ticket lifetime of 24 hours, and a password of 744275. All of the ticket lifetime values are specified in seconds.

**Known**  
The administrator has access to the KERBDFLT profile in the REALM class.

**Command**  
RALTER REALM KERBDFLT KERB(KERBNAME(KRB2000.IBM.COM)  
MINKTTLFE(300) DEFTKTLFE(36000) MAXKTTLFE(86400)  
PASSWORD(744275))

**Defaults**  
None.

Example 11  
**Operation**  
A trust relationship is being defined between the kerb390.endicott.ibm.com realm and the realm at ker2000.endicott.ibm.com.

**Known**  
The administrator has access to the /.../KERB390.ENDICOTT.IBM.COM/KRBTGT/KER2000.ENDICOTT.IBM.COM profile in the REALM class.

**Command**  
RALTER REALM /.../KERB390.ENDICOTT.IBM.COM/KRBTGT/KER2000.ENDICOTT.IBM.COM KERB(PASSWORD(12345678))

**Defaults**  
None.
RALTER

Table 82. RALTER examples (continued)

<table>
<thead>
<tr>
<th>Example</th>
<th>Operation</th>
<th>Known</th>
<th>Command</th>
<th>Defaults</th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
<td>The system default EIM values are being altered by changing the domain distinguished name and disabling it.</td>
<td>IRR.PROXY.DEFAULTS is the profile being changed in the FACILITY class. The EIM domain distinguished name begins with Pok EIM Domain, o=IBM, c=US.</td>
<td>RALTER FACILITY IRR.PROXY.DEFAULTS EIM(DOMAINDN('ibm-eimDomainName=Pok EIM Domain, o=IBM, c=US')) OPTIONS(DISABLE))</td>
<td>None.</td>
</tr>
<tr>
<td>13</td>
<td>The security administrator wants to change an attribute of the installation-defined class TSTCLAS8. He wishes to change the value of RACLIST(REQUIRED) to RACLIST(ALLOWED).</td>
<td>The administrator has the SPECIAL attribute.</td>
<td>RALTER CDT TSTCLAS8 CDTINFO(RACLIST(ALLOWED))</td>
<td>None.</td>
</tr>
<tr>
<td>14</td>
<td>At Rui’s installation, identity mappings in EIM change frequently and identity mapping changes are not refreshed often enough. She wants to reduce the MAPPINGTIMEOUT value so that mappings in the identity cache expire sooner and are refreshed more frequently from EIM. She reduces the timeout value to 1800 seconds (one-half hour).</td>
<td>When the IRR.ICTX.DEFAULTS profile was defined in the LDAPBIND class, the MAPPINGTIMEOUT value was defaulted to 3600 seconds (one hour).</td>
<td>RALTER LDAPBIND IRR.ICTX.DEFAULTS ICTX(MAPPINGTIMEOUT(1800))</td>
<td>None.</td>
</tr>
<tr>
<td>15</td>
<td>At Rui’s installation, identity mappings in EIM change frequently and identity mapping changes are not refreshed often enough. She wants to reduce the MAPPINGTIMEOUT value so that mappings in the identity cache expire sooner and are refreshed more frequently from EIM. She reduces the timeout value to 1800 seconds (one-half hour).</td>
<td>When the IRR.ICTX.DEFAULTS profile was defined in the LDAPBIND class, the MAPPINGTIMEOUT value was defaulted to 3600 seconds (one hour).</td>
<td>RALTER LDAPBIND IRR.ICTX.DEFAULTS ICTX(MAPPINGTIMEOUT(1800))</td>
<td>None.</td>
</tr>
<tr>
<td>16</td>
<td>The security administrator uses a custom field called ADDRESS in her user profiles. She wants to update the help text and modify the maximum length of this custom field.</td>
<td>The user has the SPECIAL attribute. The changes in the custom field are not effective until the system programmer rebuilds the dynamic parse table using the IRRDPI00 UPDATE command.</td>
<td>RALTER CFIELD USER.CSDATA.ADDRESS CFDEF(MAXLENGTH(200) HELP('HOME ADDRESS, 1-200 characters'))</td>
<td>None.</td>
</tr>
<tr>
<td>17</td>
<td>User SECADM wants to update the signature verification options for a controlled program called MYPROG14 program to specify that it must now be digitally signed before it can be loaded, that the program should fail to load if its digital signature cannot be verified for any reason, and that logging of signature verification events should occur only for failures.</td>
<td>The user has the SPECIAL attribute. The MYPROG14 program is a program object that resides in a partitioned data set extended (PDSE) library.</td>
<td>RALTER PROGRAM MYPROG14 SIGVER(SIGREQUIRED(YES) FAILLOAD(ANYBAD) SIGAUDIT(ANYBAD))</td>
<td>None.</td>
</tr>
</tbody>
</table>
RDEFINE (Define general resource profile)

Purpose

Use the RDEFINE command to:

- Define to RACF all resources belonging to classes specified in the class descriptor table.
- Create entries in the global access checking table.
- Define security categories and security levels.
- Define classes (as profiles in the RACGLIST class) for which RACF saves RAclistEd results on the RACF database.
- Define the attributes of classes in the dynamic class descriptor table.
- Define a custom field and its attributes.

The RDEFINE command adds a profile for the resource to the RACF database in order to control access to the resource. It also places your user ID on the access list and gives you ALTER authority to the resource unless SETROPTS NOADDCREATOR is in effect.

You cannot use the RDEFINE command to define users, groups, data sets, certificates, certificate key rings, or certificate mappings.

To have changes take effect after defining a generic profile if the class is not RAclistEd by either the SETROPTS RAclist or RACROUTE REQUEST=LIST, GLOBAL=YES, one of the following steps is required:

- The security administrator issues the SETROPTS command:

  SETROPTS GENERIC(class-name) REFRESH

  See the SETROPTS command for authorization requirements.

- The user of the resource logs off and logs on again.

To have changes take effect after defining a generic profile if the class is RAclistEd, the security administrator issues the following command:

SETROPTS RAclist(class-name) REFRESH

For more information, refer to z/OS Security Server RACF Security Administrator’s Guide.

Attention:

- When the RDEFINE command is issued from ISPF, the TSO command buffer (including SESSKEY and SSIGNON) is written to the ISPLOG data set. As a result, you should not issue this command from ISPF or you must control the ISPLOG data set carefully.
- If the RDEFINE command is issued as a RACF operator command, the command and all data is written to the system log. Therefore, use of RDEFINE as a RACF operator command should either be controlled or you should issue the command as a TSO command.

Issuing options

The following table identifies the eligible options for issuing the RDEFINE command:
As a RACF TSO command? | As a RACF operator command? | With command direction? | With automatic command direction? | From the RACF parameter library?
---|---|---|---|---
Yes | Yes | Yes | Yes | Yes

For information on issuing this command as a RACF TSO command, refer to Chapter 3, “RACF TSO commands,” on page 15.

For information on issuing this command as a RACF operator command, refer to Chapter 4, “RACF operator commands,” on page 21.

You must be logged on to the console to issue this command as a RACF operator command.

### Related commands
- To create a group profile, see ADDGROUP (Add group profile).
- To create a data set profile, see ADDSD (Add data set profile).
- To create a user profile, see ADDUSER (Add user profile).
- To permit or deny access to a general resource profile, see PERMIT (Maintain resource access lists).
- To change a general resource profile, see RALTER (Alter general resource profile).
- To delete a general resource profile, see RDELETE (Delete general resource profile).
- To obtain a list of general resource profiles, see SEARCH (Search RACF database).
- To list a general resource profile, see RLIST (List general resource profile).

### Authorization required
When issuing the RDEFINE command as a RACF operator command, you might require sufficient authority to the proper resource in the OPERCMDS class. See z/OS Security Server RACF Security Administrator’s Guide for further information.

To use the RDEFINE command, you must have either the SPECIAL attribute or minimally the CLAUTH authority for the class. If you have CLAUTH authority but not the SPECIAL attribute, you may need to be authorized as follows:
- If you have CLAUTH authority for the GLOBAL class, and group-SPECIAL authority in a group, you can add members whose high-level qualifier is the group name or a user ID in the scope of the group. This applies only to classes that are sensitive to high-level qualifiers, such as DATASET.
- If the name to be defined is not already defined to RACF as a member of a resource group and you are defining a profile in a normal (non-member, non-grouping) class, you must be authorized to define resources for the specified class. (This authority can be established with the CLAUTH operand on the ADDUSER or ALTUSER command.)
- If the name to be defined is not already defined to RACF as a member of a resource group and you are defining a profile in a member class or a member of a grouping class, you must be authorized to define resources for the specified class. (This authority can be established with the CLAUTH operand on the ADDUSER or ALTUSER command.)
RDEFINE

- If the resource to be defined is a discrete name already defined to RACF as a member of a resource group, you can define it as a resource to RACF if you have ALTER authority, or if the resource group profile is within the scope of a group in which you have the group-SPECIAL attribute, or if you are the owner of the resource group profile. If authority conflicts arise because the resource is a member of more than one group and the user's authority in those groups differs, RACF resolves the conflict by using the least restrictive authority (unless modified by the installation).
- If you define a profile in the FILE or DIRECTRY class, one of the following must be true:
  - The second qualifier of the profile name must match your user ID
  - You must have the SPECIAL attribute
  - The profile name must be within the scope of a group in which you have the group-SPECIAL attribute.
- If you do not have the SPECIAL attribute and the SETROPTS GENERICOWNER option is in effect, and if an existing generic profile protects the profile name you are defining, you need to own the less specific profile.

Notes:
1. If the less specific profile is within the scope of a group in which you have group-SPECIAL, you are considered to own the profile.
2. GENERICOWNER does not apply to the PROGRAM general resource class.
3. For additional information on the GENERICOWNER option and restricting the creation of general resource profiles, see z/OS Security Server RACF Security Administrator's Guide.

- To assign a security category to a profile, you must have the category in your user profile.
- To assign a security level to a profile, your own profile must have a security level that is equal to or greater than the security level you are defining.
- To use the ADDMEM operand, see the description of the ADDMEM operand for information on the authority required to use the operand.
- To specify the AT keyword, you must have READ authority to the DIRECT.node resource in the RRSFDATA class and a user ID association must be established between the specified node.userid pair(s).
- To specify the ONLYAT keyword you must have the SPECIAL attribute, the userid specified on the ONLYAT keyword must have the SPECIAL attribute, and a user ID association must be established between the specified node.userid pair(s) if the user IDs are not identical.
- To define segments other than the base segment, such as DLFDATA, you must have the SPECIAL attribute or your installation must permit you to do so through field-level access checking.
- To assign a security label to a profile, you must have READ access to the security label profile. However, the security administrator can limit the ability to assign security labels to only users with the SPECIAL attribute.
- Only a SPECIAL user can define a delegated resource (by specifying the RACF-DELEGATED string in the APPLDATA of the profile protecting the resource) when the resource has a SECLABEL and SETROPTS SECLABELCONTROL is in effect.
**Model profiles:** To specify a model profile (using, as required, FROM, FCLASS, FGENERIC, and FVOLUME), you must have sufficient authority over the model profile (the *from* profile). RACF makes the following checks until one of the conditions is met:

- You have the SPECIAL attribute.
- The *from* profile is within the scope of a group in which you have the group-SPECIAL attribute.
- You are the owner of the *from* profile.
- If the FCLASS operand is DATASET, the high-level qualifier of the profile name (or the qualifier supplied by the naming conventions routine or a command installation exit) is your user ID.
- For a discrete profile, you are on the access list in the *from* profile with ALTER authority. (If you have any lower level of authority, you cannot use the profile as a model.)
- For a discrete profile, your current connect group (or, if list-of-groups checking is active, any group to which you are connected) is in the access list in the *from* profile with ALTER authority.
- For a discrete profile, the universal access authority (UACC) is ALTER.

**Syntax**

For the key to the symbols used in the command syntax diagrams, see "Syntax of RACF commands and operands" on page 9. The complete syntax of the RDEFINE command is:

```plaintext
[subsystem-prefix][RDEFINE | RDEF]

class-name

(profile-name-1 ...)

[ ADDCATEGORY(category-name ...) ]
[ ADDMEM(member ...) ]
[ APPLDATA('application-data') ]
[ AT([node].userid ...) | ONLYAT([node].userid ...) ]
[ AUDIT( access-attempt [audit-access-level] ...) ]
[ CDTINFO(
    [ CASE( UPPER | ASIS ) ]
    [ DEFAULTRC( 0 | 4 | 8 ) ]
    [ DEFAULTUACC( A|CEE | ALTER | CONTROL
                            | UPDATE | READ | NONE ) ]
    [ FIRST( characters-allowed ... ) ]
    [ GENERIC( ALLOWED | DISALLOWED ) ]
    [ GENLIST( ALLOWED | DISALLOWED ) ]
    [ GROUP( grouping-class-name ) ]
    [ KEYQUALIFIERS( 0 | nnn ) ]
    [ MACPROCESSING( NORMAL | REVERSE | EQUAL ) ]
    [ MAXLENGTH( 8 | nnn ) ]
    [ MAXLENX( nnn ) ]
    [ MEMBER( member-class-name ) ]
    [ OPERATIONS( YES | NO ) ]
    [ OTHER( characters-allowed ... ) ]
    [ POSIT( nnn ) ]
    [ PROFILEALLOWED( YES | NO ) ]
    [ RACLST( ALLOWED | DISALLOWED | REQUIRED ) ]
    [ SECLABELSREQUIRED( YES | NO ) ]
    [ SIGNAL( YES | NO ) ]
)]
```

RDEFINE
RDEFINE

[ CFDEF(
    [ TYPE( CHAR | FLAG | HEX | NUM ) ]
    [ FIRST( ALPHA | ALPHANUM | ANY
        | NONATABC | NONATNUM | NUMERIC ) ]
    [ HELP( help-text ) ]
    [ LISTHEAD( list-heading-text ) ]
    [ MAXLENGTH( maximum-field-length ) ]
    [ MAXVALUE( maximum-numeric-value ) ]
    [ MINVALUE( minimum-numeric-value ) ]
    [ MIXED( YES | NO ) ]
    [ OTHER( ALPHA | ALPHANUM | ANY
        | NONATABC | NONATNUM | NUMERIC ) ]
  ) ]

[ DATA('installation-defined-data') ]

[ DLFDATA(
    [ RETAIN( YES | NO ) ]
    [ JOBNAMES(jobname-1 ... ]
   ) ]

[ EIM(
    [ DOMAINDN(eim_domain_dn) ]
    [ OPTIONS( ENABLE | DISABLE ) ]
    [ LOCALREGISTRY(registry_name) ]
    [ KERBREGISTRY(registry_name) ]
    [ X509REGISTRY(registry_name) ]
  ) ]

[ FCLASS(profile-name-2-class) ]

[ FGENERIC ]

[ FROM(profile-name-2) ]

[ FVOLUME(profile-name-2-serial) ]

[ ICSF(
    [ ASYMUSAGE(
        [ HANDSHAKE | NOHANDSHAKE ]
        [ SECUREEXPORT | NOSECUREEXPORT ]
      ) ]
    [ SYMEXPORTABLE(BYANY | BYLIST | BYNONE) ]
    [ SYMEXPORTCERTS([qualifier]label-name ... | *) ]
    [ SYMEXPORTKEYS(ICSF-key-label ... | *) ]
  ) ]

[ ICTX(
    [ USEMAP( YES | NO ) ]
    [ DOMAP( YES | NO ) ]
    [ MAPREQUIRED( YES | NO ) ]
    [ MAPPINGTIMEOUT(nnnn) ]
  ) ]

[ KERB(
    [ DEFTKTLFE(def-ticket-life) ]
    [ ENCRYPT(
        [ DES | NODES ]
        [ DES3 | NODES3 ]
        [ DESD | NODES3D ]
        [ AES128 | NODES128 ]
        [ AES256 | NODES256 ]
      ) ]
    [ KERBNAME(kerberos-realm-name) ]
    [ MAXTKTLFE(max-ticket-life) ]
    [ MINTKTLFE(min-ticket-life) ]
    [ PASSWORD(kerberos-password) ]
  ) ]

[ LEVEL(nn) ]
RDEFINE

```
[ NOTIFY(userid) ]
[ OWNER(userid or group-name) ]
[ PROXY(
    [ LDAPHOST(ldap_url) ]
    [ BINDDN(bind_distinguished_name) ]
    [ BINDPW(bind_password) ]
) ]
[ SECLABEL(seclabel-name) ]
[ SECLEVEL(seclabel-name) ]
[ SESSION(
    [ CONVSEC( NONE | CONV | ALREADYV | PERSISTV | AVPV ) ]
    [ INTERVAL(n) ]
    [ LOCK ]
    [ SESSKEY(session-key) ]
) ]
[ SIGVER(
    [ SIGREQUIRED( YES | NO ) ]
    [ FAILLOAD( ANYBAD | BADSIGONLY | NEVER ) ]
    [ SIGAUDIT( ALL | SUCCESS | ANYBAD | BADSIGONLY | NONE ) ]
) ]
[ SINGLEDSN ]
[ SSIGNON(
    [ KEYMASKED(key-value)
    | KEYENCRYPTED(key-value) ]
) ]
[ STDATA(
    [ USER(userid | =MEMBER) ]
    [ GROUP(group-name | =MEMBER) ]
    [ PRIVILEGED( YES | NO ) ]
    [ TRACE( YES | NO ) ]
    [ TRUSTED( YES | NO ) ]
) ]
[ SVFMR(
    [ SCRIPTNAME(script-name) ]
    [ PARMNAME(parm-name) ]
) ]
[ TIMEZONE( {E | W} hh.mm ) ]
[ TME(
    [ CHILDREN(profile-name ... ) ]
    [ GROUPS(group-name ...) ]
    [ PARENT(profile-name) ]
    [ RESOURCE(resource-access-specification ... ) ]
    [ ROLES(role-access-specification ... ) ]
) ]
[ TVTOC ]
[ UACC(access-authority) ]
[ WARNING ]
[ WHEN([DAYS(day-info) ] [TIME(time-info) ] ) ]
```

For information on issuing this command as a RACF TSO command, refer to Chapter 3, “RACF TSO commands,” on page 15.

For information on issuing this command as a RACF operator command, refer to Chapter 4, “RACF operator commands,” on page 21.
Parameters

**subsystem-prefix**
Specifies that the RACF subsystem is the processing environment of the command. The subsystem prefix can be either the installation-defined prefix for RACF (1–8 characters) or, if no prefix has been defined, the RACF subsystem name followed by a blank. If the command prefix was registered with CPF, you can use the MVS command D OPDATA to display it or you can contact your RACF security administrator.

Only specify the subsystem prefix when issuing this command as a RACF operator command. The subsystem prefix is required when issuing RACF operator commands.

**class-name**
Specifies the name of the class to which the resource belongs. The valid class names are those defined in the class descriptor table. For a list of general resource classes defined in the class descriptor table supplied by IBM, see Appendix B, “Supplied RACF resource classes,” on page 659.

This operand is required and must be the first operand following RDEFINE.

This command is not intended to be used for profiles in the following classes:
- DCEUUDS
- DIGTCERT
- DIGTNMAP
- DIGTRING
- IDIDMAP
- NDSSLINK
- NOTELINK
- ROLE
- UNIXMAP.

**profile-name-1**
Specifies the name of the discrete or generic profile you want to add to the specified class. RACF uses the class descriptor table to determine if the class is defined to RACF, the syntax of resource names within the class, and whether the resource is a group resource. For more information, see Appendix A, “Naming considerations for resource profiles” and z/OS Security Server RACF Security Administrator’s Guide.

Mixed-case profile names are accepted and preserved when class-name refers to a class defined in the static class descriptor table with CASE=ASIS or in the dynamic class descriptor table with CASE(ASIS).

This operand is required and must be the second operand following RDEFINE.
- If you specify more than one profile name, you must enclose the list of names in parentheses.
- In general, you should not specify profile names within single quotation marks because most classes will not allow this, and the RDEFINE command will fail. Classes such as FACILITY (or others whose class definition allows any character as the first character) will allow RDEFINE to work, but this will result in defining a profile whose name contains the single quotation mark. As
RDEFINE

a result, during authorization checking, the profile might not protect the resources intended to be protected. In fact, such a profile can only work if you also have a resource manager that encloses its resource names in single quotation marks, but most resource managers do not.

- If you specify class-name as GLOBAL, profile-name-1 must be either DATASET or a valid class name (other than a resource group class) as specified in the class descriptor table. If you specify class-name as GLOBAL or SECDATA and also specify ADDMEM, you can specify only one profile name.

- If you want RACF to store the results from a SETROPTS RACLST or a RACROUTE REQUEST=LIST,GLOBAL=YES in the RACGLIST class, define the base profile for the class by issuing the following RDEFINE command, where profile-name-1 is a valid class in the class descriptor table:

  RDEFINE RACGLIST profile-name-1

If the RACGLIST class is active when the class profile-name-1 is RACLSTed, RACF stores the RACLSTed results as profile-name-1_nnnnn profiles on the RACF database. For example, the following RDEFINE command creates a base profile DASDVOL:

Example:

RDEFINE RACGLIST DASDVOL

The following SETROPTS command stores the RACLST results as profiles DASDVOL_00001, DASDVOL_00002, and so on, in the RACGLIST class.

Example:

SETROPTS RACLST(DASDVOL)

Restrictions: The following classes can not be specified as profile names for RACGLIST:

- The CDT, GLOBAL, RACGLIST, USER, CONNECT, GROUP, and DATASET classes
- Any class noted in Appendix B, “Supplied RACF resource classes,” on page 659 as not allowing profiles in the class (for example, the DIRAUTH class)
- Any class noted in Appendix B, “Supplied RACF resource classes,” on page 659 as not intended for use with any RACF command (for example, the SCDMBR class)
- Any resource grouping class, other than the NODES and RACFVARS grouping classes which are allowed.

- If class-name is a resource grouping class (other than NODES or RACFVARS), you cannot specify a generic profile-name-1. If class-name is DLFCLASS, you should not specify a generic profile-name-1 as it is ignored by DLF processing.

- If you specify class-name as PROGRAM, you can specify only one profile name, and you must specify the ADDMEM operand.

- If you specify class-name as PROGRAM, profile-name must be the name of a load module. If you specify the full name of the load module, the profile applies only to that module. If you specify the last character of the name as an asterisk (*), the profile applies to all load modules that match the preceding part of the name, and these load modules must all reside in the same library. For example, IKF* identifies all load module names that begin...
with IKF. If you specify profile-name as an asterisk (*), the profile applies to all load modules that reside in the library you identify on the ADDMEM operand.

- If you are activating field-level access checking, you must specify class-name as FIELD. To define a profile (profile-name-1) in the FIELD class, you must follow the profile naming conventions described in “Field-level access checking” in z/OS Security Server RACF Security Administrator’s Guide.

- If you specify class-name as STARTED, you must specify two qualifiers for the profile name. Follow the profile naming conventions described in “Specifying STARTED Class Profile Names” in z/OS Security Server RACF Security Administrator’s Guide.

- If you specify class-name as CFIELD, you must follow the profile naming conventions described in “Profiles in the CFIELD class” in z/OS Security Server RACF Security Administrator’s Guide.

Notes:
1. Do not specify a generic character unless SETROPTS GENERIC (or SETROPTS GENCMD) is in effect.
2. RACF processes each resource you specify independently, and all operands you specify apply to each named resource. If an error occurs while it is processing a resource, RACF issues a message and continues processing with the next resource.

**ADDCATEGORY(category-name ...)**
Specifies one or more names of installation-defined security categories. The names you specify must be defined as members of the CATEGORY profile in the SECDATA class. (For information on defining security categories, see z/OS Security Server RACF Security Administrator’s Guide)

When the SECDATA class is active and you specify ADDCATEGORY, RACF performs security category checking in addition to its other authorization checking. If a user requests access to a resource, RACF compares the list of security categories in the user’s profile with the list of security categories in the resource profile. If RACF finds any security category in the resource profile that is not in the user’s profile, RACF denies access to the resource. If the user’s profile contains all the required security categories, RACF continues with other authorization checking.

*Note:* RACF does not perform security category checking for a started task with the RACF privileged or trusted attribute. The RACF privileged or trusted attribute can be assigned to a started task through the RACF started procedures table or STARTED class. Also, RACF does not enforce security category information specified on profiles in the PROGRAM class.

**ADDMEM(member ...)**
Specifies the member names that RACF is to add to the profile indicated by profile-name-1. The meaning of member varies, depending on the class.

You can use the ADDMEM operand to perform tasks such as defining security categories and security levels, entries in the global access checking table, and entries for program control, or to implement security labels on a system basis, as described in the following sections.

Mixed-case member names are accepted and preserved when class-name refers to a class defined in the static class descriptor table with CASE=ASIS or in the dynamic class descriptor table with CASE(ASIS). When class-name is GLOBAL and profile-name is the name of a class defined in the static class
**RDEFINE**

descriptor table with CASE=ASIS or in the dynamic class descriptor table with CASE(ASIS), the name part of a member entry in the GLOBAL access table is preserved as entered.

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

RDEF GIMS GIMSGRP ADDMEM(ABC*)

and you are looking for a specific member, so you enter:

RLIST TIMS ABCD RESGROUP

The GIMS profile GIMSGRP will not appear in the output.

**Note:** When considering this example, if you are unable to define the profile ABCD, it might be due to a generic definition somewhere in GIMS.

For ADDMEM with the DATASET class, no characters including generic characters, such as the asterisk (+) and the percent sign (%), can be combined with the value &RACUID to form a single qualifier level of the member name. This restriction does not exist for ADDMEM with classes other than DATASET.

For ADDMEM with the RACFVARS class, generic characters, such as the ampersand (&), the asterisk (+) and the percent sign (%) should not be specified in the member names.

In addition to the authority needed to issue the RDEFINE command, you need one of the following authorities to add members using the RDEFINE command:

1. For classes other than SECLABEL, PROGRAM, SECDATA, GLOBAL, RACFVARS, and NODES, if the member resources are already RACF-protected by a member class profile or as a member of a profile in the same grouping class, one of the following must be true:
   - You have ALTER access authority to the member.
   - You are the owner of the member resource.
   - The member resource is within the scope of a group in which you have the group-SPECIAL attribute.
   - You have the SPECIAL attribute.

2. For classes other than SECLABEL, PROGRAM, SECDATA, GLOBAL, RACFVARS, and NODES, if the member resources are not RACF-protected (that is, there is no profile defined for that member), one of the following must be true:
   - You have CLAUTH authority to define resources in the member resource class.
   - You have the SPECIAL attribute.

3. To add a member to a profile in the RACFVARS or NODES class, one of the following must be true:
   - You have CLAUTH authority to define resources in the specified class (for example, RACFVARS or NODES).
   - You have the SPECIAL attribute.
   - You are the owner of the profile indicated by profile-name-1.
   - You have ALTER access authority to the profile indicated by profile-name-1.

4. To add a member to a profile in the SECLABEL, PROGRAM or SECDATA class, one of the following must be true:
You have CLAUTH authority to define resources in the specified class (for example, PROGRAM or SECDATA).

You have the SPECIAL attribute.

5. To add a member to a profile in the GLOBAL class (other than the GLOBAL DATASET, GLOBAL DIRECTRY, or GLOBAL FILE profile) using the following syntax:

```
RDEF GLOBAL class-name
    ADDMEM(resource-name/access-level)
```

If the profile `resource-name` is already RACF-protected by a profile in class `class-name`, one of the following must be true:

- You have ALTER access authority to the profile `resource-name` in class `class-name`.
- You are the OWNER of the profile `resource-name`.
- The profile `resource-name` in class `class-name` is within the scope of a group in which you have the group-SPECIAL attribute.
- You have the SPECIAL attribute.

6. If the profile `resource-name` is not already RACF-protected (that is, there is no profile defined for that member in class `class-name`):

- You have CLAUTH authority to define resources in the class `class-name`.
- You have the SPECIAL attribute.

7. To add a member to the GLOBAL DATASET profile, one of the following must be true:

- You are the owner of the DATASET profile in the GLOBAL class.
- The member is within the scope of a group in which you have the group-SPECIAL attribute, or the high-level qualifier of the member name is your user ID.
- You have the SPECIAL attribute.

8. To add a member to the GLOBAL DIRECTRY or GLOBAL FILE profile, you must have the SPECIAL attribute.

RACF ignores the ADDMEM operand if the class name you specify is not a resource grouping class, SECLABEL, GLOBAL, SECDATA, NODES, or PROGRAM.

**Specifying member on the ADDMEM operand**

The following sections describe how to specify members for each of the following classes:

- Resource grouping classes
- SECLABEL
- GLOBAL
- SECDATA
- NODES
- PROGRAM.

The descriptions for these classes are below.

- **When a resource grouping class is the class name**

  *Resource Grouping Class:* If the class name is a resource grouping class, the members you specify through the ADDMEM operand protects the resources in the related member class.
If generic profile checking is active for the related member class, you can include a generic character (*, **, &, or % only) in the member to protect multiple resources.

For more information on resource grouping classes and their related member classes, see [z/OS Security Server RACF Security Administrator’s Guide](#).

- **When SECLABEL is the class name**

  Security Label By System: You can define a security label for use on specific systems. Issue the RDEFINE command specifying the system identifier (SMFID) of the system on which the security label can be used. Note that RACF-defined SECLABELs (SYSHIGH, SYSLOW, SYSNONE, and SYSMULTI) are not affected by SECLABEL by System.

  The format of this command is as follows:

  ```
  RDEFINE SECLABEL profile-name
  ADDMEM(system-identifier)
  ```

  The `system-identifier` is the 4-character value specified for the SID parameter of the SMFPRMxx member of SYS1.PARMLIB. See [z/OS MVS Initialization and Tuning Reference](#) for additional information on SMFPRMxx. RACF does not check that the specified `system-identifier` actually exists in SMFPRMxx.

  The security label will only be restricted to the systems specified by ADDMEM if the SETR SECLBYSYSTEM option is active. If this option is not active, or ADDMEM is not specified, the security label can be used on all systems. Changes to profiles in the SECLABEL class are activated by issuing SETR RACLIST(SECLABEL) REFRESH.

- **When GLOBAL is the class name**

  Global Access Checking: You can define an entry in the global access checking table by issuing the RDEFINE command with the following operands:

  - GLOBAL as the `class-name`
  - The appropriate resource class name as `profile-name`
  - ADDMEM with the name of the entry you are defining (as `member`). (If the name you specify as `member` contains a generic character (+ or %), generic profile checking (SETROPTS command with the GENERIC operand) must be active for the resource class you specify as `profile-name`.)
  - The access level you are assigning to the entry (member) using the following format:

    ```
    member[/{ALTER|CONTROL|NONE|READ|UPDATE}]
    ```

  The format of this command is as follows:

  ```
  RDEFINE GLOBAL profile-name
  ADDMEM(member/access-level)
  ```

  Each entry you define controls global access checking for the resources matching that entry name.

  **Important**

  Because RACF performs global access checking before security classification processing, might allow access to a resource you are protecting with a security category, security level, or both. To avoid a security exposure to a sensitive resource, do not define an entry in the global access checking table for a resource you are protecting with security classification processing.
When you define an entry in the global access checking table, specify `member` on the ADDMEM operand as described in the following sections.

- **Global access checking for data sets**
  When you define an entry in the global access checking table for a data set, enclose the entry name in single quotation marks if you do not want your TSO prefix (which might be your user ID) used as the high-level qualifier of the entry name.

  For example, assume that your user ID is SMITH. If you issue the following command, you define the entry SMITH.ABC in the global access table.

  ```plaintext
  RDEFINE GLOBAL DATASET ADDMEM('SMITH.ABC'/READ)
  ```

  If you do not enclose the entry name in single quotation marks, your TSO prefix is used as the high-level qualifier of the entry name. For example, if you issue the following command, you define the entry SMITH.ABC in the global access table.

  ```plaintext
  RDEFINE GLOBAL DATASET ADDMEM(ABC/READ)
  ```

  If the entry name you specify contains `*` as the high-level qualifier and you do not enclose the name in single quotation marks, RACF creates the entry exactly as you specify it (your TSO prefix is not used as the high-level qualifier of the entry name). For example, if you issue the following command, you define the entry `*.*ABC` in the global access table. If you enclose `*.*ABC` in single quotation marks, you define the same entry (`*.*ABC`) in the global access table.

  ```plaintext
  RDEFINE GLOBAL DATASET ADDMEM(*.*ABC/READ)
  ```

- **Global access checking for general resources**
  To define an entry in the global access checking table for a general resource, specify any valid class name in the class descriptor table as a profile name. (For a list of general resource classes defined in the class descriptor table supplied by IBM, see Appendix B, “Supplied RACF resource classes,” on page 659.) The member name you specify with the ADDMEM operand can contain one or more generic characters (%, *, or **). For information on using generic characters, see Appendix A, "Naming considerations for resource profiles." 

- **When SECDATA is the class name**
  Security Classification of Users and Data: To define a security category or security level for your installation, specify `class-name` as SECDATA and `profile-name` as one of the following:
  - CATEGORY when defining a security category
  - SECLEVEL when defining a security level.

  If you specify SECDATA CATEGORY, the ADDMEM operand specifies the name of an installation-defined category of users.

  For example, to define three categories of users named CODE, TEST, and DOC, issue:

  ```plaintext
  RDEFINE SECDATA CATEGORY ADDMEM(CODE TEST DOC)
  ```

  If you specify SECDATA SECLEVEL, the ADDMEM operand specifies both the name of an installation-defined security level and the number you assign to that level, in the form:

  ```plaintext
  seclvel-name/seclvel-number
  ```

  You must separate the two items by a forward slash character (/). The `seclvel-name` can contain 1–44 characters and must not contain a blank,
comma, semicolon, or right parenthesis. The *seclevel-number* can be any number 1–254. The higher the number, the higher the security level. For example, to define three security levels, where CONFIDENTIAL is the most restrictive, enter:

```
RDEFINE SECDATA SECLEVEL
  ADDMEM(GENERAL/10 EXPERIMENTAL/75 CONFIDENTIAL/150)
```

Because RACF keeps track of security levels by number, replacing an existing security level name does not affect the protection that the security level number provides. If you had defined the security levels shown in the preceding example and then replaced GENERAL/10 with INTERNAL/10, a listing of a user or resource profile that included security level 10 would show the new name. Because the security level number is the same, there is no need to change any resource or user profiles.

When you actually change an existing CATEGORY profile or SECLEVEL profile, however, RACF issues a warning message to remind you that the change is not reflected in existing resource or user profiles. In this case, you can use the SEARCH command to locate the profiles you must modify.

- **When NODES is the class name**

  Specify only one value with the ADDMEM operand. If you specify multiple values, RACF stores them in the NODES profile but translates using only the last one specified.

  **Restriction:** When more than one value is defined in a NODES profile, you cannot use the RLIST command to determine which value was the last one specified.

  **Guideline:** If one or more values are already defined in a NODES profile, use the DELMEM operand to remove them before specifying the new value.


  **Translation of User IDs, Group Name, or Security Labels on Inbound Jobs or SYSOUT:**

  If the class name is NODES, you can specify how user IDs, group names, and security labels are translated. The translation depends on the second and third qualifiers of the profile name, as follows:

<table>
<thead>
<tr>
<th>If the second qualifier is...</th>
<th>The ADDMEM value specifies...</th>
</tr>
</thead>
<tbody>
<tr>
<td>RUSER</td>
<td>The user ID to be used on this system for the jobs originating from NJE nodes to which the profile applies</td>
</tr>
<tr>
<td>USERJ</td>
<td>The user ID to be used on this system for the inbound jobs to which the profile applies</td>
</tr>
<tr>
<td>USERS</td>
<td>The user ID to be used on this system for the inbound SYSOUT to which the profile applies</td>
</tr>
<tr>
<td>GROUPJ</td>
<td>The group name to be used on this system for the inbound jobs to which the profile applies</td>
</tr>
<tr>
<td>GROUPS</td>
<td>The group name to be used on this system for the inbound SYSOUT to which the profile applies</td>
</tr>
<tr>
<td>SECLJ</td>
<td>The security label to be used on this system for the inbound jobs to which the profile applies</td>
</tr>
<tr>
<td>SECLS</td>
<td>The security label to be used on this system for the inbound SYSOUT to which the profile applies</td>
</tr>
</tbody>
</table>

- **When PROGRAM is the class name**
Program Control: If you specify class-name as PROGRAM, profile-name must identify one or more controlled programs (load modules or program objects), and member identifies the library containing the programs, the volume serial of that library, and a processing option. Additionally, APPLDATA may contain information that RACF will process. You specify the member entry in the following format: library-name/volume-serial/PADCHK or NOPADCHK

library-name
Specifies the name of the library in which the controlled programs reside. If profile-name is * or **, RACF treats all load modules in the specified library as controlled programs.

If it is necessary to define a program that resides in the system’s LPA or dynamic LPA as a controlled program (for example: to give it the MAIN or BASIC attribute), define the program with a profile-name that does not end in *, specify 'LPALST' as the library name, and omit the volume serial.

When it is necessary to define a specific profile for a program in the LPA, 'LPALST' should be used as the library name and the volume serial should be omitted.

The following represent valid ADDMEM values for program XYZ which exists in one of the LPA libraries or in the dynamic LPA:

- 'LPALST'
- 'LPALST'//PADCHK
- 'LPALST'//NOPADCHK

volume-serial (optional)
Specifies the serial number of the volume on which the library resides. You can use six asterisks within single quotation marks to specify the current SYSRES volume: library-name/'******'/PADCHK or NOPADCHK.

Notes:
1. The '******' value works when the SYSRES resides on more than one volume, but it applies only when the data set lives on the IPL volume.
2. If volume-serial is not specified, the specified library can exist on any volume. The alternate formats are:

   library-name//NOPADCHK
   or
   library-name

PADCHK | NOPADCHK
Specifies that RACF is to make (PADCHK) or not to make (NOPADCHK) the checks for program-accessed data sets when a user is executing the controlled programs. If you specify PADCHK, RACF verifies that (1) the conditional access list in the profile for a program-accessed data set allows the access and (2) no task in the user’s address space has previously loaded a non-controlled program.

If you specify NOPADCHK, RACF does not perform this extra checking to verify that a non-controlled program cannot access a program-accessed data set. NOPADCHK allows you, for example, to define entire libraries of modules (such as ISPF) as controlled programs without then having to grant each of these modules access to many program-accessed data sets.  

"Examples" on page 510 show two ways to
RDEFINE

define controlled programs. Before defining or modifying PROGRAM profiles, see the program control sections of z/OS Security Server RACF Security Administrator’s Guide.

APPLDATA('application-data')

Specifies a text string that is associated with each of the named resources. The text string can contain a maximum of 255 characters and must be enclosed in single quotation marks. It can also contain double-byte character set (DBCS) data.

Rules:

- For profiles in the PROGRAM class, RACF will examine the APPLDATA value (if any) and perform special processing if you have specified MAIN or BASIC (optionally followed by blanks).
  - This processing will occur only for profiles whose names do not end in *, and only when you have enabled enhanced PGMSECURITY mode.
  - For details of this processing, see z/OS Security Server RACF Security Administrator’s Guide.
- For the FACILITY class, RACF examines the APPLDATA value of the following profiles:
  - BPX.UNIQUE.USER
    The APPLDATA value specifies the name of a user profile from which RACF can copy OMVS segment information (other than UID) when assigning unique UIDs through a callable service.
  - BPX.DEFAULT.USER
    The APPLDATA value specifies a user ID and group name from which RACF can retrieve default OMVS segment information. Beginning with z/OS Version 1 Release 11, the BPX.DEFAULT.USER profile is ignored when the BPX.UNIQUE.USER profile is defined.
  - BPX.NEXT.USER
    The APPLDATA value specifies information that RACF will use for the automatic assignment of OMVS UIDs and GIDs.
  - IRR.PGMSECURITY
    The APPLDATA value specifies whether RACF will operate in basic, enhanced, or enhanced-warning PGMSECURITY mode.
    - If the APPLDATA value contains the string ENHANCED, then RACF will run in enhanced PGMSECURITY mode.
    - If the APPLDATA value contains the string BASIC, then RACF will run in basic PGMSECURITY mode.
    - If the APPLDATA is empty or contains any other value, RACF will run in enhanced PGMSECURITY mode but in warning mode rather than failure mode.
  - IRR.PROGRAM.SIGNING.group.userid
  - IRR.PROGRAM.SIGNING.userid
  - IRR.PROGRAM.SIGNING.group
  - IRR.PROGRAM.SIGNING
    For any of the IRR.PROGRAM.SIGNING profiles, the APPLDATA value specifies the signing hash algorithm, and the SAF key ring to use when signing a program.
  - IRR.PROGRAM.SIGNATURE.VERIFICATION
    The APPLDATA value specifies the SAF key ring to use when verifying the signature of a signed program.
RDEFINE

- For the TIMS and GIMS class, specify application-data as REVERIFY to force the user to reenter his password whenever the transaction or transactions listed in the profile-name or ADDMEM operands are used.

- For the PTKTDATA class, the application data field can be used to control the replay protection function of PassTicket support.
  - PassTicket replay protection prevents the use of user IDs to be shared among multiple users. However, in some events it is desirable to bypass this replay protection function.
  - Specifying no replay protection in the application data field indicates that replay protection is to be bypassed. For example, the following command would successfully result in replay protection being bypassed.

  ```
  RDEFINE PTKTDATA profile-name
  APPLDATA('NO REPLAY PROTECTION')
  ```

  Note the following:
  - There must be a single space between the words no and replay, and between replay and protection. Lack of spaces, or additional spaces or characters, will make the command ineffective. For example, entering the following command would not result in replay protection being bypassed.

  ```
  RDEFINE PTKTDATA profile-name
  APPLDATA('NOREPLAY PROTECTION')
  ```
  - The text string no replay protection will always be translated to uppercase.
  - The text string no replay protection can appear anywhere in the APPLDATA field.

- For the APPL class, when the APPLDATA value contains the RACF-INITSTATS(DAILY) string, RACF records statistics only for the first user verification of the day for the applications protected by this profile. The RACF-INITSTATS(DAILY) string is reserved text and may appear anywhere in the APPLDATA field. For more information about statistics collection, see [z/OS Security Server RACF Security Administrator’s Guide](https://www.ibm.com/support/knowledgecenter/SSLTBK_3.2.0/com.ibm.zos.zos.racf/...)

- Specifying the RACF-DELEGATED string in the APPLDATA designates the resources protected by the profile as delegated, meaning that RACROUTE REQUEST=FASTAUTH should honor a nested ACEE during access checking to this resource. The RACF-DELEGATED string is reserved text and may appear anywhere in the APPLDATA field. For more information on nested ACEEs and delegated resources, see [z/OS Security Server RACF Security Administrator’s Guide](https://www.ibm.com/support/knowledgecenter/SSLTBK_3.2.0/com.ibm.zos.zos.racf/...)

RACF does not validate the APPLDATA value during RALTER. Depending on the function, RACF might or might not issue any messages during subsequent processing if it finds an unexpected value.

The APPLDATA value, if present, can be displayed with the RLIST command.

For detailed information about each APPLDATA value, see [z/OS Security Server RACF Security Administrator’s Guide](https://www.ibm.com/support/knowledgecenter/SSLTBK_3.2.0/com.ibm.zos.zos.racf/...)

AT | ONLYAT

The AT and ONLYAT keywords are only valid when the command is issued as a RACF TSO command.
RDEFINE

AT([node].userid ...)  
Specifies that the command is to be directed to the node specified by node,  
where it runs under the authority of the user specified by userid in the  
RACF subsystem address space.  
If node is not specified, the command is directed to the local node.

ONLYAT([node].userid ...)  
Specifies that the command is to be directed only to the node specified by node where it runs under the authority of the user specified by userid in the RACF subsystem address space.  
If node is not specified, the command is directed only to the local node.

AUDIT(access-attempt[(audit-access-level)])  
Specifies which access attempts and access levels you want logged to the SMF data set.  
access-attempt  
Specifies which access attempts you want logged to the SMF data set. The following options are available:  
ALL  
Specifies that you want to log both authorized accesses and detected unauthorized access attempts.  
FAILURES  
Specifies that you want to log detected unauthorized access attempts. This is the default value if you do not specify access-attempt.  
NONE  
Specifies that you do not want any logging to be done.  
SUCCESS  
Specifies that you want to log authorized accesses to the resource.  

audit-access-level  
Specifies which access levels you want logged to the SMF data set. The levels you can specify are:  
ALTER  
Logs ALTER access-level attempts only.  
CONTROL  
Logs access attempts at the CONTROL and ALTER levels.  
READ  
Logs access attempts at any level. This is the default value if no access level is specified.  
UPDATE  
Logs access attempts at the UPDATE, CONTROL, and ALTER levels.  

FAILURES(READ) is the default value if the AUDIT operand is omitted from the command.  
You cannot audit access attempts for the EXECUTE level.

CDTINFO  
Specifies information used in the definition of an installation-defined class in the dynamic class descriptor table (CDT). For details about defining classes in the dynamic CDT, see [Administering the Dynamic Class Descriptor Table (CDT)] in z/OS Security Server RACF Security Administrator's Guide.  

Note: CDTINFO should only be specified for profiles in the CDT class.

CASE ( UPPER | ASIS )  
Specifies whether mixed-case profile names are allowed for the class.  
ASIS  
When ASIS is specified, RACF commands preserve the
RDEFINE

RDEFINE

case of profile names for the specified class. Lowercase characters are allowed in any position of the profile name where alphabetic characters are allowed, based on the character restrictions in the FIRST and OTHER keywords.

UPPER

When UPPER is specified, RACF translates the profile names for the specified class to uppercase. If CASE is not specified, CASE(UPPER) is the default.

DEFAULTTRC

Specifies the return code that RACF will provide from RACROUTE REQUEST=AUTH or REQUEST=FASTAUTH when both RACF and the class are active and (if required) the class has been processed using SETROPTS RACLST; but RACF doesn’t find a profile to protect the resource specified on the AUTH or FASTAUTH request. The return codes are interpreted as follows:

0  The access request was accepted.

4  No profile exists.

8  The access request was denied.

If DEFAULTTRC is not specified, DEFAULTTRC(4) is the default.

DEFAULTUACC

DEFAULTUACC ( ALTER | CONTROL | UPDATE | READ | NONE )

Specifies the minimum access allowed if the access level is not set when a resource profile is defined in the class.

DEFAULTUACC ( ACEE )

If no universal access level is specified at the time the profile is created, RACF uses the default universal access authority from the command issuer’s ACEE, as specified on the UACC operand of the ADDUSER, ALTUSER or CONNECT command.

If DEFAULTUACC is not specified, DEFAULTUACC(NONE) is the default.

FIRST (characters-allowed ...)

Specifies a character type restriction for the first character of the profile name. One or more of the following may be specified.

• ALPHA—Allows an alphabetic character (A–Z)

• NUMERIC—Allows a digit (0–9)

• NATIONAL—Allows characters # (X’7B’), @ (X’7C’), and $ (X’5B’)

• SPECIAL—Allows any character except the following:
  – a blank
  – a comma
  – a parenthesis
  – a semicolon
  – those characters in ALPHA, NUMERIC, or NATIONAL.

If FIRST is not specified, FIRST(ALPHA, NATIONAL) is the default.

GENERIC ( ALLOWED | DISALLOWED )

Specifies whether or not SETROPTS GENERIC and SETROPTS GENCMD are allowed for the class. The SETROPTS GENERIC command activates generic profile checking for a class. The SETROPTS GENCMD command activates generic profile command processing.
RDEFINE

If GENERIC is not specified, GENERIC(ALLOWED) is the default. If GENERIC(DISALLOWED) is specified, GENLIST(ALLOWED) cannot also be specified.

Because generic processing is not allowed for grouping classes, GENERIC(DISALLOWED) should be specified if MEMBER(member-class-name) is also specified. If GENERIC(ALLOWED) is specified or defaulted for a grouping class, a warning message is issued. Subsequent processing for the dynamic class being defined and for profiles in that class will be treated as if GENERIC(DISALLOWED) was specified.

**Rule:** If the dynamic class you are defining shares a POSIT number with other classes, all classes with the shared POSIT number must have the same GENERIC keyword value. This is because the SETROPTS GENERIC and SETROPTS GENCMD commands process all classes that share a POSIT number. If at least one class specifies GENERIC(DISALLOWED) and at least one class specifies GENERIC(ALLOWED), RACF issues a warning message. When you subsequently add this class to the dynamic class descriptor table using the SETROPTS RACLIST(CDT) command, RACF might change the value of the GENERIC keyword to match the GENERIC keyword value of the other classes sharing the POSIT number.

- If this dynamic class shares a POSIT number with an IBM-supplied class, RACF changes the value of the GENERIC keyword in the dynamic class to match the IBM class. (The class attribute in the IBM-supplied class takes precedence).
- If this dynamic class shares a POSIT number with an installation-defined class (static or dynamic), RACF determines the least restrictive attribute—GENERIC(ALLOWED) is less restrictive than GENERIC(DISALLOWED)— and changes the GENERIC(DISALLOWED) class attribute to GENERIC(ALLOWED).

**Exception:** A grouping class and member class can share a POSIT number although their GENERIC keyword values need not match. You must specify GENERIC(DISALLOWED) for the grouping class. However, you can specify either ALLOWED or DISALLOWED for the member class.

**GENLIST ( ALLOWED | DISALLOWED )**

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

If GENLIST is not specified, GENLIST(DISALLOWED) is the default.

**GROUP ( grouping-class-name )**

Specifies the name of the class that groups the resources within the specified class. If GROUP is not specified, RACF does not allow resource grouping for the class. The grouping-class-name must be 1–8 characters.

When GROUP is specified, the class being defined is a member class.

If GROUP is specified, then grouping-class-name must also be defined in the CDT class and its MEMBER keyword should refer to the class being defined. The GROUP and MEMBER keywords must have matching class entries before SETROPTS RACLIST(CDT) is issued to build or refresh the

478 z/OS V1R11.0 Security Server RACF Command Language Reference
KEYQUALIFIERS ( 0 | nnn )
Specifies the number of matching qualifiers RACF uses when loading genericprofilenames to satisfy an authorization request if a discrete profile does not exist for a resource. For example, if you specify two for the class, all generic profile names whose highest level qualifiers match the two highest qualifiers of the entity name are loaded into the user's storage when the user requests access to a resource. The nnn value must be a number 0–123.

If KEYQUALIFIERS is not specified, the default is 0 and profile names for the entire class are loaded and searched.

The maximum value you can specify is 123, which is the maximum number of qualifiers in a name 246 characters long.

When KEYQUALIFIERS(nnn) is specified, generic profiles created in that class may not contain generic characters in the first nnn qualifiers of the profile name.

When KEYQUALIFIERS(nnn) is greater than 0 for a class, all discrete and generic profiles in that class must have at least nnn+1 qualifiers in each profile name. The number of qualifiers a profile name is determined by counting the number of period characters in the profile and adding one; the first character is not examined.

Examples of valid profile names for KEYQUALIFIERS(2) are:
A.B.C
A.B.**
A.B.C.0*

Guideline: Specify KEYQUALIFIERS(nnn) greater than 0 for classes that have the following characteristics:
- The class is not usually RACLISTed or GENLISTed.
- Profile names in the class follow a naming convention where many generic profiles have the same nnn qualifiers at the beginning of the profile name.

For example, if you have an application that uses an installation-defined class to protect reports on terminal usage, you might have profiles such as these for each user on your z/OS system:

REPORTS.USER1.TERMUSE.*
REPORTS.USER1.TERMUSE.DEPT60.*
REPORTS.USER1.TERMUSE.2006.JAN.*
REPORTS.USER1.TERMUSE.2006.FEB.*
REPORTS.USER1.TERMUSE.2006.MAR.*
REPORTS.USER1.TERMUSE.2006.APR.*
REPORTS.USER1.TERMUSE.2006.MAY.*
REPORTS.USER1.TERMUSE.2006.JUN.*
REPORTS.USER1.TERMUSE.2006.JUL.*
REPORTS.USER1.TERMUSE.2006.AUG.*
REPORTS.USER1.TERMUSE.2006.SEP.*
REPORTS.USER1.TERMUSE.2006.OCT.*
REPORTS.USER1.TERMUSE.2006.NOV.*
REPORTS.USER1.TERMUSE.2006.DEC.*

In this example, you might define your installation class using KEYQUALIFIERS(3) so that when RACF checks authorization checks for resources in your class, only generic profile names that match the first three qualifiers of your report are loaded into storage for RACF to check.
Restriction: Different rules apply for the FILE and DIRECTORY classes. For the syntax required for profile names in the DIRECTORY and FILE classes, see the appropriate RACF Command Language Reference for your VM system.

MACPROCESSING (NORMAL | REVERSE | EQUAL)
Specifies which type of mandatory access control (MAC) processing is required for the class. If MACPROCESSING is not specified, MACPROCESSING(NORMAL) is the default.

- NORMAL—specifies normal MAC processing is required. If and when a MAC check is performed, the user’s SECLABEL must dominate that of the resource.
- REVERSE—specifies reverse MAC processing is required. If and when a MAC check is performed, the SECLABEL of the resource must dominate that of the user.
- EQUAL—specifies equal MAC processing is required. If and when a MAC check is performed, the SECLABEL of the user must be equivalent to that of the resource. MACPROCESSING(EQUAL) should be used for classes where two-way communication is expected. Writedown (SETROPTS MLS) does not apply to classes where MACPROCESSING(EQUAL) is specified.

MAXLENGTH (8 | nnn)
Specifies the maximum length of resource and profile names for the specified class when MAXLENX is not specified. When MAXLENX is also specified, MAXLENGTH represents the maximum length of a resource name only when a RACROUTE macro is invoked with the ENTITY keyword. The value of nnn must be 1–246.

If MAXLENGTH is not specified, the default is 8.

MAXLENX (nnn)
Specifies the maximum length of resource and profile names for the specified class when a RACROUTE macro is invoked with the ENTITYX keyword, or when a profile is added or changed using a RACF command processor. The value of nnn value must be 1–246.

If MAXLENX is not specified before SETROPTS RAclist(CDT) is issued to build or refresh the dynamic CDT or before the system is restarted, the value specified for MAXLENGTH is used for MAXLENX in subsequent processing for the dynamic class.

MEMBER (member-class-name)
Specifies the name of the class grouped by the resources within the specified class. The member-class-name must be 1–8 characters.

When MEMBER is specified, the class being defined is a resource group.

If MEMBER is specified, then member-class-name must also be defined in the CDT class and its GROUP keyword should refer to the class being defined. The GROUP and MEMBER keywords must have matching class entries before SETROPTS RAclist(CDT) is issued to build or refresh the dynamic CDT or before the system is restarted; otherwise, the class in error will not be added to the dynamic class descriptor table.

OPERATIONS (YES | NO)
Specifies whether RACF is to take the OPERATIONS attribute into account when it performs authorization checking. If YES is specified, RACF considers the OPERATIONS attribute; if NO is specified, RACF ignores the OPERATIONS attribute.
If OPERATIONS is not specified, OPERATIONS(NO) is the default.

**OTHER** *(characters-allowed ...)*

Specifies a character type restriction for the characters of the profile name other than the first character. One or more of the following may be specified:

- **ALPHA**—Allows an alphabetic character (A–Z)
- **NUMERIC**—Allows a digit (0–9)
- **NATIONAL**—Allows characters # (X'7B'), @ (X'7C'), and $ (X'5B')
- **SPECIAL**—Allows any character except the following:
  - a blank
  - a comma
  - a parenthesis
  - a semicolon
  - those characters in ALPHA, NUMERIC, or NATIONAL.

If OTHER is not specified, OTHER(ALPHA, NATIONAL) is the default.

**POSIT** *(nnn)*

Specifies the POSIT number associated with the class. Each class in the class descriptor table has a POSIT number specified which identifies a set of option flags that control the following RACF processing options:

- Whether authorization checking should take place for the class (SETROPTS CLASSACT).
- Whether auditing should take place for resources within the class (SETROPTS AUDIT).
- Whether statistics should be kept for resources within the class (SETROPTS STATISTICS).
- Whether generic profile access checking is active for the class (SETROPTS GENERIC).
- Whether generic command processing is active for the class (SETROPTS GENCMD).
- Whether global access checking is active for the class (SETROPTS GLOBAL).
- Whether the user has CLAUTH to a resource class.
- Whether special resource access auditing applies to the class (SETROPTS LOGOPTIONS).
- Whether SETROPTS RACLIST will occur for this class (when RACLST(ALLOWED) or RACLST(REQUIRED) is also specified).
- For all classes that have the same POSIT number specified, these options are identical. If you change an option for one class, this change will also affect all other classes that share the same POSIT number.

Before you issue SETROPTS RACLIST(CDT) to build or refresh the dynamic class descriptor table, you must decide whether to use a unique set of option flags for each RACF class or whether to have two or more RACF classes share the same set of option flags. If you choose to use a unique set of option flags for a class, assign the class a unique POSIT number. If you choose to share the same set of option flags among several classes, assign those classes the same POSIT number.

Before you issue SETROPTS RACLIST(CDT), the POSIT keyword must specify a valid value on the RDEFINE command. Otherwise, the new class will not be added to the dynamic class descriptor table.
Once you issue SETROPTS RACLIST(CDT) to build or refresh the dynamic class descriptor table, you can activate the classes that comprise it and their respective set of option flags by using the appropriate keywords on the SETROPTS command.

There are 1024 POSIT numbers that can identify 1024 sets of option flags. Installations can specify POSIT numbers 19–56 and 128–527. POSIT numbers 0–18, 57–127 and 528–1023 are reserved for IBM use and should not be specified for installation-defined classes unless an installation intends that one of its classes share SETROPTS options with an IBM-defined class.

**Guideline:** A RACF class that has a default return code 8 should *not* share a POSIT value with a RACF class having a different default return code. If a class with a default return code 8 is activated but no profiles are defined, user activity that requires access in that class will be prevented.

**PROFILESALLOWED ( YES | NO )**

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

If PROFILESALLOWED is not specified, PROFILESALLOWED(YES) is the default.

**RACLIST**

Specifies whether SETROPTS RACLIST is to be allowed, disallowed or required for the specified class. If you process this class using SETROPTS RACLIST, RACF brings copies of all discrete and generic profiles within that class into storage in a data space. RACF uses those profiles in storage to check the authorization of any users who want to access the resources. The profiles remain in storage until removed by SETROPTS NORACLIST.

**ALLOWED**

Specifies that SETROPTS RACLIST may be used for the class, but is not required for authorization checking.

**DISALLOWED**

Specifies that SETROPTS RACLIST may not be used for the class.

**REQUIRED**

Specifies that you must process the class using SETROPTS RACLIST in order to use RACROUTE REQUEST/Auth. The purpose of this keyword is to allow routines that cannot tolerate I/O to invoke RACF. When this keyword is specified and the class is not processed by SETROPTS RACLIST and a RACROUTE REQUEST=AUTH is attempted, the return code is 4.

If RACLIST is not specified, RACLIST(DISALLOWED) is the default.

**SECLABELSREQUIRED ( YES | NO )**

Specifies whether a SECLABEL is required for the profiles of the specified class when SETROPTS MLACTIVE is on.

SECLABELSREQUIRED(NO) means that RACF will not require a SECLABEL for profiles in this class; however, if a SECLABEL exists for this profile and the SECLABEL class is active, RACF will use it during authorization checking. SECLABELSREQUIRED(NO) applies to general
resource classes that have no profiles, such as DIRAUTH, or for classes that contain no data, such as OPERCMDS and SECLABEL.

SECLABELSREQUIRED(YES) means that RACF will require a SECLABEL for profiles in this class when SETROPTS MLACTIVE is on.

If SECLABELSREQUIRED is not specified, SECLABELSREQUIRED(NO) is the default.

**SIGNAL ( YES | NO )**

Specifies whether an ENF signal should be sent to listeners when RACLISTed profiles are created, updated or deleted for authorization checking.

When SIGNAL(YES) is specified, RACF will send an ENF signal to listeners when a SETROPTS RACLIST, SETROPTS NORACLIST or a SETROPTS RACLIST REFRESH is issued for the class to activate, deactivate or update the profiles used for authorization checking. For more information, see the [SIGNAL section of z/OS Security Server RACF System Programmer’s Guide](https://www.ibm.com/hurs/jettelz/).

When SIGNAL(NO) is specified, no ENF signal is sent.

SIGNAL(YES) is not valid if RACLIST(DISALLOWED) is specified.

If SIGNAL is not specified, SIGNAL(NO) is the default.

**CFDEF**

Defines a custom field for profiles in the CFIELD class, and specifies the name and attributes for the custom field. The custom fields you define with the CFDEF operand can be used in the CSDATA segment of user and group profiles. For more information about custom fields, including the profile name format, see [z/OS Security Server RACF Security Administrator’s Guide](https://www.ibm.com/hurs/jettelz/).

New custom fields are not effective until the system programmer rebuilds the dynamic parse table using the IRRDPI00 UPDATE command. For information about using the IRRDPI00 command, see [z/OS Security Server RACF System Programmer’s Guide](https://www.ibm.com/hurs/jettelz/).

**Rule**: Specify CFDEF only for profiles in the CFIELD class.

**TYPE**

Specifies the data type of the custom field. If you do not specify TYPE, CHAR is the default.

For each data type, you can restrict the content of the custom field using the attributes shown in Table 83 on page 484. For each attribute shown, the default values based on data type, are as follows.

**Rule**: For each data type, do not specify attributes noted in Table 83 on page 484 with a dash (—).
Table 83. Default values for attributes that restrict the content of a custom field, based on data type.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Default value based on the TYPE attribute</th>
</tr>
</thead>
<tbody>
<tr>
<td>FIRST</td>
<td>CHAR: ALPHA</td>
</tr>
<tr>
<td>MAXLENGTH</td>
<td>1100</td>
</tr>
<tr>
<td>MAXVALUE</td>
<td>—</td>
</tr>
<tr>
<td>MINVALUE</td>
<td>—</td>
</tr>
<tr>
<td>MIXED</td>
<td>NO</td>
</tr>
<tr>
<td>OTHER</td>
<td>CHAR: ALPHA</td>
</tr>
</tbody>
</table>

**Note:** If you do not specify MAXVALUE with TYPE(NUM), it defaults to the length of the highest value based on the MAXLENGTH value.

**CHAR**
Specifies that the custom field is a character field.

**Guidelines:**
- When you specify TYPE(CHAR), specify values for the following attributes:
  - FIRST: The default value is ALPHA.
  - MAXLENGTH: The default value is 1100.
  - MIXED: The default value is NO.
  - OTHER: The default value is ALPHA.
- To allow a custom field value to be specified as a quoted string, specify FIRST(ANY) and OTHER(ANY).

**Rule:** Do not specify MAXVALUE or MINVALUE with TYPE(CHAR).

**FLAG**
Specifies that the custom field is a flag field.

**Rule:** Do not specify any attributes with TYPE(FLAG). The default values are sufficient and required. The default values are FIRST(NONATABC), OTHER(NONATABC), and MAXLENGTH(3).

**HEX**
Specifies that the custom field is a hexadecimal field.

**Guideline:** When you specify TYPE(HEX), specify a value for the MAXLENGTH attribute. The default value is 512. Specify an even number because hexadecimal data is stored and displayed as an even number of characters.

**Rule:** Do not specify FIRST, OTHER, MAXVALUE, MINVALUE, or MIXED with TYPE(HEX).

**NUM**
Specifies that the custom field is a numeric field.

**Guidelines:**
- When you specify TYPE(NUM), specify values for the MAXVALUE and MINVALUE attributes.
- You need not specify MAXLENGTH with TYPE(NUM) because MAXVALUE limits the numeric value.

**Rule:** Do not specify FIRST, OTHER, or MIXED with TYPE(NUM).
FIRST
Specifies a character restriction for the first character in the custom field.

Guideline: Do not specify FIRST for custom fields with FLAG, HEX, or NUM data type. If you incorrectly specify the FIRST value for the data type, the custom field might be unusable.

Rules: The valid options for the FIRST attribute apply as follows, based on TYPE value (data type).

<table>
<thead>
<tr>
<th>Valid options</th>
<th>CHAR</th>
<th>FLAG</th>
<th>HEX</th>
<th>NUM</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALPHA</td>
<td>✔</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ALPHANUM</td>
<td>✔</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ANY</td>
<td>✔</td>
<td>✔</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NONATABC</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>NONATNUM</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>NUMERIC</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
</tbody>
</table>

For each option of the FIRST attribute, the characters allowed in the custom field are as follows:

<table>
<thead>
<tr>
<th>Valid options</th>
<th>Alphabetic characters (A–Z)</th>
<th>National characters (X’7B’, X’7C’, X’5B’)</th>
<th>Numeric characters (0–9)</th>
<th>Any other character</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALPHA</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>ALPHANUM</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>ANY</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>NONATABC</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>NONATNUM</td>
<td>✔</td>
<td></td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>NUMERIC</td>
<td>✔</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

ALPHA
Allows alphabetic characters (A–Z) and national characters # (X’7B’), @ (X’7C’), and $ (X’5B’).

ALPHANUM
Allows alphabetic characters (A–Z), numbers (0–9), and national characters # (X’7B’), @ (X’7C’), and $ (X’5B’).

ANY
Allows alphabetic characters (A–Z), numbers (0–9), national characters # (X’7B’), @ (X’7C’), and $ (X’5B’), and any other character. When you specify both FIRST(ANY) and OTHER(ANY), also allows quoted strings.

NONATABC
Allows alphabetic characters, and excludes numbers and national characters # (X’7B’), @ (X’7C’), and $ (X’5B’).
RDEFINE

NONATNUM

Allows alphabetic characters and numbers, but excludes national characters # (X'7B'), @ (X'7C'), and $ (X'5B')

NUMERIC

Allows numbers (0–9).

If you do not specify FIRST, the values default as follows, based on TYPE value (data type).

<table>
<thead>
<tr>
<th>Data type</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>CHAR</td>
<td>ALPHA</td>
</tr>
<tr>
<td>FLAG</td>
<td>NONATABC</td>
</tr>
<tr>
<td>HEX</td>
<td>NONATNUM</td>
</tr>
<tr>
<td>NUM</td>
<td>NUMERIC</td>
</tr>
</tbody>
</table>

HELP( help-text )

Specifies the help text for this custom field. The help text is displayed when the user is in TSO PROMPT mode and presses the PF1 key or enters a question mark (?). Lowercase alphabetic characters in the help-text value are translated to upper case.

Rules:
- Length: 1–255 characters.
- If the help text contains parentheses, commas, blanks, or semicolons, enclose the entire text string in single quotation marks.
- If a single quotation mark is intended to be part of the help text, use two single quotation marks together for each single quotation mark within the string, and enclose the entire string in single quotation marks.

Example: To define help text for a customer's address and indicate that the field can be up to 100 characters, you might specify the following value:

HELP(‘CUSTOMER’S ADDRESS. SPECIFY UP TO 100 CHARACTERS’)

If you do not specify HELP, the value defaults to the custom field name defined in CFIELD profile name.

LISTHEAD( list-heading-text )

Specifies the heading to display in the output for the LISTUSER or LISTGRP command whenever the CSDATA segment is listed. Lowercase alphabetic characters in the list-heading-text value are translated to upper case.

Rules:
- Length: 1–40 characters.
- If the heading text contains parentheses, commas, blanks, or semicolons, enclose the entire text string in single quotation marks.
- If a single quotation mark is intended to be part of the help text, use two single quotation marks together for each single quotation mark within the string, and enclose the entire string in single quotation marks.

Example:

LISTHEAD(‘CUSTOMER’S ADDRESS =’)

Guidelines: If you specify a LISTHEAD value, avoid confusion for users who use the LISTUSER or LISTGRP command to list custom field values by following these guidelines:
- Ensure that each custom field has a unique heading.
Append an equal sign (=) or other delimiter to your LISTHEAD value to indicate in the list output where the heading ends and the data begins.

If you do not specify LISTHEAD, the value defaults to the custom field name defined in CFIELD profile name and an equal sign (=) is appended to the end of the value.

**MAXLENGTH( maximum-field-length )**

Specifies the maximum length of the custom field. You can specify MAXLENGTH with any TYPE value (data type).

**Guideline:** Do not specify with TYPE(FLAG) because 3 is the default value and the only valid value.

**Rules:** The valid values or value ranges shown in Table 84 apply based on data type.

<table>
<thead>
<tr>
<th>Data type</th>
<th>Valid value or range</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>CHAR</td>
<td>1–1100</td>
<td>1100</td>
</tr>
<tr>
<td>FLAG</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>HEX</td>
<td>1–512</td>
<td>512</td>
</tr>
<tr>
<td>NUM</td>
<td>1–10</td>
<td>10</td>
</tr>
</tbody>
</table>

If you do not specify MAXLENGTH, the default values shown in Table 84 apply based on data type.

**MAXVALUE( maximum-numeric-value )**

Specifies the maximum numeric value for a custom field with TYPE(NUM).

**Rules:**
- Valid range: 0–2 147 483 647
- Do not specify a MAXVALUE value for custom fields with CHAR, FLAG, or HEX data type.
- Do not specify a MAXVALUE value lower than the MINVALUE value.
- Do not specify a MAXVALUE value longer than the highest value based on MAXLENGTH value.

If you do not specify MAXVALUE, the value defaults to the length of the highest value based on the MAXLENGTH value. For example, if you specify MAXLENGTH(4), the default MAXVALUE is 9999.

**MINVALUE( minimum-numeric-value )**

Specifies the minimum numeric value for a custom field with TYPE(NUM).

**Rules:**
- Valid range: 0–2 147 483 647
- Do not specify a MINVALUE value for fields with CHAR, FLAG, or HEX data type.
- Do not specify a MINVALUE value higher than the MAXVALUE value.
- Do not specify a MINVALUE value longer than the highest value based on MAXLENGTH value.

If you do not specify MINVALUE, the value defaults to 0.

**MIXED( YES | NO )**

Specifies whether mixed-case alphabetic characters are allowed for a custom field with TYPE(CHAR).
RDEFINE

YES  
Lowercase characters are allowed in any position of the custom field where alphabetic characters are allowed, based on the character restrictions specified with the FIRST and OTHER attributes. RACF commands, such as ADDUSER, do not translate lowercase alphabetic characters in the field to upper case.

Rule: Do not specify MIXED(YES) for custom fields with FLAG, HEX, or NUM data type.

NO  
RACF commands translate lowercase alphabetic characters in the field to upper case.

If you do not specify MIXED, the value defaults to NO.

OTHER  
Specifies a character restriction for characters in the custom field other than the first character.

Guideline: Do not specify OTHER for custom fields with FLAG, HEX, or NUM data type. If you incorrectly specify the OTHER value for the data type, the custom field might be unusable.

For each option of the OTHER attribute, the characters allowed in the custom field are as follows:

<table>
<thead>
<tr>
<th>Valid options</th>
<th>Characters allowed</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Alphabetic characters (A–Z)</td>
</tr>
<tr>
<td>ALPHA</td>
<td>✓</td>
</tr>
<tr>
<td>ALPHANUM</td>
<td>✓</td>
</tr>
<tr>
<td>ANY</td>
<td>✓</td>
</tr>
<tr>
<td>NONATABC</td>
<td>✓</td>
</tr>
<tr>
<td>NONATNUM</td>
<td>✓</td>
</tr>
<tr>
<td>NUMERIC</td>
<td>✓</td>
</tr>
</tbody>
</table>

ALPHA  
Allows alphabetic characters (A–Z) and national characters # (X’7B’), @ (X’7C’), and $ (X’5B’).

ALPHANUM  
Allows alphabetic characters (A–Z), numbers (0–9), and national characters # (X’7B’), @ (X’7C’), and $ (X’5B’).

ANY  
Allows alphabetic characters (A–Z), numbers (0–9), national characters # (X’7B’), @ (X’7C’), and $ (X’5B’), and any other character. When you specify both FIRST(ANY) and OTHER(ANY), also allows quoted strings.

NONATABC  
Allows alphabetic characters, and excludes numbers and national characters # (X’7B’), @ (X’7C’), and $ (X’5B’).

NONATNUM  
Allows alphabetic characters and numbers, but excludes national characters # (X’7B’), @ (X’7C’), and $ (X’5B’).
NUMERIC

Allows numbers (0–9).

**Rules:** The valid options for the OTHER attribute apply as follows, based on TYPE value (data type).

<table>
<thead>
<tr>
<th>Valid options</th>
<th>Data type based on TYPE attribute</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>CHAR</td>
</tr>
<tr>
<td>ALPHA</td>
<td>✓</td>
</tr>
<tr>
<td>ALPHANUM</td>
<td>✓</td>
</tr>
<tr>
<td>ANY</td>
<td>✓</td>
</tr>
<tr>
<td>NONATAABC</td>
<td>✓</td>
</tr>
<tr>
<td>NONATNUM</td>
<td>✓</td>
</tr>
<tr>
<td>NUMERIC</td>
<td>✓</td>
</tr>
</tbody>
</table>

If you do not specify OTHER, the values default as follows, based on TYPE value (data type).

<table>
<thead>
<tr>
<th>Data type</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>CHAR</td>
<td>ALPHA</td>
</tr>
<tr>
<td>FLAG</td>
<td>NONATAABC</td>
</tr>
<tr>
<td>HEX</td>
<td>NONATNUM</td>
</tr>
<tr>
<td>NUM</td>
<td>NUMERIC</td>
</tr>
</tbody>
</table>

**DATA('installation-defined-data')**

Specifies up to 255 characters of installation-defined data to be stored in the profile for the resource and the data must be enclosed in single quotation marks. It can also contain double-byte character set (DBCS) data.

This information is listed by the RLIST command.

**DLFDATA**

Specifies information used in the control of DLF objects in profiles in the DLFCLASS.

**RETAI N(YES | NO)**

Specifies whether the DLF object can be retained after use.

**JOBNAMES(jobname-1 ...)**

Specifies the list of objects which can access the DLF objects protected by this profile.

You can specify any job name valid on your system. You can also specify generic job names with an asterisk (*) as the last character of the job name. For example, JOBNAMES(ABC) allows only job ABC to access the DLF objects protected by the profile. JOBNAMES(ABC*) allows any job whose name begins with ABC (such as ABC, ABC1, or ABCDEF and so forth) to access the DLF objects.

If DLFDATA is not specified, or is specified without the RETAIN suboperand, RETAIN(NO) is defaulted.

**EIM**

The EIM and PROXY segment keywords and subkeywords combine to define the EIM domain, the LDAP host it resides on, and the bind information required by the EIM services to establish a connection with an EIM domain. The EIM services will attempt to retrieve this information when it is not explicitly supplied with the invocation parameters.
RDEFINE

**DOMAINDN**(eim_domain_dn)
Specifies the distinguished name of the EIM domain. A valid EIM domain distinguished name begins with `ibm-eimDomainName=`. Uppercase and lowercase characters are accepted and maintained in the case in which they are entered. The EIM domain distinguished name is one component of an EIM domain name.

An EIM domain name identifies the LDAP server that stores the EIM domain information. The EIM domain name begins with the `ldap_url` from the LDAPHOST suboperand of the PROXY keyword, followed by `/` and ends with the `eim_domain_dn` from the DOMAINDN suboperand. The length of a valid EIM domain name is determined by the combination of those factors. RACF allows the input of 1023 characters for the domain distinguished name. RACF does not ensure that an EIM domain name created from the LDAP URL and EIM domain distinguished name forms a valid EIM domain name.

For more information about LDAP distinguished names, see *IBM Tivoli Directory Server Administration and Use for z/OS* or *z/OS Integrated Security Services LDAP Server Administration and Use*.

**OPTIONS**
Specifies options that control the EIM configuration.

**ENABLE | DISABLE**

*ENABLE* Specifies that new connections may be established with the specified EIM domain. This is the default.

*DISABLE* Specifies that new connections may not be established with the specified EIM domain.

**LOCALREGISTRY**(registry_name)
Specifies the name of the local RACF registry in EIM domains. This operand is valid only with the following profiles and is ignored for all others:

- The IRR.PROXY.DEFAULTS profile in the FACILITY class
- The IRR.ICTX.DEFAULTS.sysid profile in the LDAPBIND class
- The IRR.ICTX.DEFAULTS profile in the LDAPBIND class.

EIM uses the `registry_name` value defined in the IRR.PROXY.DEFAULTS profile. The ICTX identity cache `registry_name` uses the value defined in the IRR.ICTX.DEFAULTS.sysid or IRR.ICTX.DEFAULTS profile.

The `registry_name` value is 1–255 characters in length. It can consist of any characters and can be entered with or without single quotation marks. The following rules apply:

- If parentheses, commas, blanks, or semicolons are intended as part of the `registry_name`, you must enclose the entire character string in single quotation marks.
- If a single quotation mark is intended as part of the `registry_name`, use two single quotation marks together for each single quotation mark within the string, and enclose the entire string within single quotation marks.
- Both uppercase and lowercase characters are accepted and maintained in the case in which they are entered.

**KERBREGISTRY**(registry_name)
Specifies the name of the Kerberos registry in the EIM domain that the
**RDEFINE**

system is configured to use. This operand is only valid for the IRR.PROXY.DEFAULTS FACILITY class profile. The value is ignored when used on other profiles.

The Kerberos registry_name may be 1–255 characters long. Uppercase and lowercase characters are allowed, but are not significant because the Kerberos registry name is stored in the RACF database in uppercase.

**X509REGISTRY(registry_name)**

Specifies the name of the X.509 registry in the EIM domain that the system is configured to use. This operand is only valid for the IRR.PROXY.DEFAULTS FACILITY class profile. The value is ignored when used on other profiles.

The X.509 registry_name may be 1–255 characters long. Uppercase and lowercase characters are allowed, but are not significant because the X.509 registry name is stored in the RACF database in uppercase.

**FCLASS(profile-name-2-class)**

Specifies the name of the class to which profile-name-2 belongs. The valid class names are DATASET and those classes defined in the class descriptor table. For a list of general resource classes defined in the class descriptor table supplied by IBM, see Appendix B, “Supplied RACF resource classes,” on page 659.

If you omit this operand, RACF assumes that profile-name-2 belongs to the same class as profile-name-1. This operand is valid only when you also specify the FROM operand; otherwise, RACF ignores it.

**FGENERIC**

Specifies that RACF is to treat profile-name-2 as a generic name, even if it is fully qualified (meaning that it does not contain any generic characters). This operand is needed only if profile-name-2 is a DATASET profile.

**FROM(profile-name-2)**

Specifies the name of an existing discrete or generic profile that RACF is to use as a model for the new profile. The model profile name you specify on the FROM operand overrides any model name specified in your user or group profile. If you specify FROM and omit FCLASS, RACF assumes that profile-name-2 is the name of a profile in the same class as profile-name-1.

Mixed-case profile names are accepted and preserved when FCLASS refers to a class defined in the static class descriptor table with CASE=ASIS or in the dynamic class descriptor table with CASE(ASIS).

To specify FROM, you must have sufficient authority to both profile-name-1 and profile-name-2, as described under “Authorization Required”.

**Possible Changes to Copied Profiles When Modeling Occurs**

When a profile is copied during profile modeling, the new profile might differ from the model in the following ways:

- Certain conditional access list conditions are valid only for specific classes. For example, WHEN(SYSID) is only valid for the PROGRAM class and WHEN(PROGRAM) is only valid for data sets and the SERVAUTH class. PROGRAM and SYSID entries in the conditional access list of profile-name-2 will be copied to the conditional access list of profile-name-1 only if the condition is valid for the class of profile-name-1.
- RACF places the user’s ID on the access list with ALTER access authority or, if the user’s ID is already on the access list, RACF changes the user’s
RDEFINE

access authority to ALTER. However, if NOADDCREATOR is in effect, RACF
copies the access list authorities exactly as they appear in the model's
access list.

- If the model profile contains members (specified with the ADDMEM operand),
  the members are not copied into the new profile.

- If the SETROPTS MLS option is in effect, the security label (if specified in
  the model profile) is not copied. Instead, the user's current security label is
  used.

  Exception: When SETROPTS MLS and MLSTABLE are both in effect and
  the user has the SPECIAL attribute, the security label specified in the model
  profile is copied to the new profile.

- For TAPEVOL profiles, TVTOC information is not copied to the new profile.

- Information in the non-RACF segments (for example, the SESSION or
  DLFDATA segment) is not copied.

For information about automatic profile modeling, refer to z/OS Security Server
RACF Security Administrator's Guide.

FVOLUME(volume-serial)

Specifies the volume RACF is to use to locate the model profile
(profile-name-2).

If you specify FVOLUME and RACF does not find profile-name-2 associated
with that volume, the command fails. If you omit this operand and
profile-name-2 appears more than once in the RACF data set, the command
fails.

FVOLUME is valid only when FCLASS either specifies or defaults to DATASET
and when profile-name-2 specifies a discrete profile. Otherwise, RACF ignores
FVOLUME.

ICSF

Specifies ICSF attributes for the keys that are controlled by this profile. ICSF
attributes are valid only for profiles in the CSFKEYS, GCSFKEYS, XCSFKEY,
and GXCSFKEY classes.

ASYMUSAGE

Specifies how an asymmetric key that is controlled by this profile is eligible
to be used. If you do not specify ASYMUSAGE, the key is eligible for all
uses.

SECUREEXPORT | NOSECUREEXPORT

  Specifies whether the key is eligible to be used to export or import
  symmetric keys.

HANDSHAKE | NOHANDSHAKE

  Specifies whether the key is eligible to be used to protect
  communication channels.

SYMEXPORTABLE

Specifies which public keys, if any, are eligible for use to export a
symmetric key that is controlled by this profile. If you do not specify
SYMEXPORTABLE, any public key is eligible.

BYANY

  Any public key is eligible. The SYMEXPORTCERTS and
  SYMEXPORTKEYS settings are ignored. This option is the default
  setting.
**RDEFINE**

**BYLIST**
Only public keys specified with the SYMEXPORTCERTS or SYMEXPORTKEYS option are eligible. If neither option is set for this symmetric key, no public key is eligible (as if BYNONE were specified).

**BYNONE**
No public key is eligible. The SYMEXPORTCERTS and SYMEXPORTKEYS settings are ignored.

**SYMEXPORTCERTS([qualifier]/label-name ... | *)**
Specifies a list of the labels of digital certificates that are eligible to be used to export the symmetric keys controlled by this profile.

Each listed certificate must exist in the ICSF key store (the SAF key ring or PKCS #11 token specified by an ICSF configuration setting). For information about the ICSF key store, see `z/OS Cryptographic Services ICSF Administrator's Guide`.

Specify an asterisk (*) to indicate that any certificate in the ICSF key store is eligible to be used to export the symmetric keys controlled by this profile. Specifying an asterisk (*) overrides any listed labels.

Specify each certificate label using a certificate label string in the form of `qualifier/label-name`.

**qualifier**
Specifies an optional qualifier in the certificate label string when multiple certificates have the same label. If specified, RACF translates the qualifier value to uppercase characters before storing it in the profile. The meaning of the qualifier value depends on where the certificate resides.

<table>
<thead>
<tr>
<th>When the certificate resides in a ...</th>
<th>The qualifier value is ...</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAF key ring</td>
<td>The RACF user ID of the certificate owner.</td>
</tr>
<tr>
<td>PKCS #11 token</td>
<td>The value of the CKA_ID attribute of the certificate. The CKA_ID value consists of up to 64 hexadecimal characters. Valid characters are 0–9 and A–F.</td>
</tr>
</tbody>
</table>

**/label-name**
Specifies the certificate label assigned when the certificate was created. You must specify the forward slash character (/) followed by the certificate label.

If the certificate label contains blanks, or special characters that cause problems with TSO/E, such as the comma, parenthesis, or comment delimiter (/*), the entire certificate label string must be enclosed in single quotation marks.

Any leading or trailing blanks specified in `label-name` are removed from this value before storing it in the profile.

**Examples of certificate label strings:**
- DENICE/CertForDenice
- 'ROGERS/Cert for Rogers'
- '/DLR cert'

**SYMEXPORTKEYS(ICSF-key-label ... | *)**
Specifies a list of the ICSF key labels of public keys that are eligible to be
used to export the symmetric keys controlled by this profile. Each listed
public key must reside in the ICSF PKA key data set (PKDS).

Specify an asterisk (*) to indicate that any public key in the ICSF PKDS is
eligible to be used to export the symmetric keys controlled by this profile.
Specifying an asterisk (*) overrides any listed labels.

**ICSF-key-label**

Specifies the ICSF key label for the public key. The label name cannot
exceed 64 characters. The first character must be an alphabetic
color or a national character ($, $, or $). Subsequent characters
can be a period character (.) or any alphanumeric or national character.

**ICTX**

Specifies the ICTX configuration options that control the ICTX identity cache.
The ICTX identity cache uses an in-storage copy of the configuration options.
Use SETROPTS RACLIST processing for the LDAPBIND class to activate
these options. (See the z/OS Security Server RACF Security Administrator’s
Guide for more information about SETROPTS RACLIST processing.)

For details about the ICTX configuration options, see z/OS Integrated Security
Services EIM Guide and Reference.

The following operands are used only for the following profiles in the
LDAPBIND class and are ignored for other profiles:

- IRR.ICTX.DEFAULTS.sysid
- IRR.ICTX.DEFAULTS

**USEMAP (YES | NO)**

Specifies whether the ICTX identity cache stores a valid identity mapping to
a z/OS user ID when provided by the application. If you do not specify this,
USEMAP(YES) is the default.

**YES** When the application provides a valid mapping to a local z/OS user
ID, the ICTX identity cache stores it. (This is the default value.)

**NO** Identity mappings provided by the application are not stored.

**DOMAP (YES | NO)**

Specifies whether the ICTX identity cache uses Enterprise Identity Mapping
(EIM) services to find a mapping to a z/OS user ID for an authenticated
user, and then stores the mapping. If you do not specify this, DOMAP(NO)
is the default.

**YES** When EIM finds a mapping to a z/OS user ID for an authenticated
user, the ICTX identity cache stores it.

**NO** The ICTX identity cache will not use EIM to find an identity
mapping. (This is the default value.)

**MAPREQUIRED(YES | NO)**

Specifies whether the ICTX identity cache requires identity mapping to a
z/OS user ID for an authenticated user. If you do not specify this,
MAPREQUIRED(NO) is the default.

**YES** The ICTX identity cache fails the request when no valid mapping is
provided by the application or found using EIM.

**NO** The ICTX identity cache does not fail the request when no valid
mapping is provided by the application or found using EIM.
MAPPINGTIMEOUT(1–3600)
   Specifies how long (one second to one hour) the ICTX identity cache stores
   an identity mapping to a z/OS user ID for an authenticated user. If you do
   not specify this, MAPPINGTIMEOUT(3600) is the default.

   Guideline: If you frequently modify your EIM mappings, consider a low
   MAPPINGTIMEOUT value. A shorter timeout period causes the ICTX
   identity cache to invoke EIM more frequently. This allows your cached
   mappings to be refreshed more frequently and improves their currency.

KERB
   Specifies z/OS Integrated Security Services Network Authentication Service
   information for a REALM class profile.

DEFTKTLFE(def-ticket-life)
   Specifies the default ticket lifetime for the local z/OS Network Authentication
   Service in seconds. The value of DEFTKTLFE is 1–2 147 483 647. Note
   that 0 is not a valid value.

   This keyword is only applicable when defining the KERBDFLT REALM
   profile for the local realm.

   If DEFTKTLFE is specified, MAXTKTLFE and MINTKTLFE must also be
   specified.

ENCRIPT
   Specifies which keys can be used by the z/OS Network Authentication
   Service realm you are defining.

   ENCRYPT is the default value when you specify KERB. The default values
   for ENCRYPT are DES, DES3, DESD, AES128, and AES256.

   DES | NODES
      Whether DES encrypted keys can be used.

   DES3 | NODES3
      Whether DES3 encrypted keys can be used.

   DESD | NODES3
      Whether DESD encrypted keys can be used.

   AES128 | NOAES128
      Whether AES128 encrypted keys can be used.

   AES256 | NOAES256
      Whether AES256 encrypted keys can be used.

   When a realm's password changes, a key of each type is generated and
   stored in the principal's user profile. The use of each key is based on the
   z/OS Network Authentication Service configuration.

   See z/OS Integrated Security Services Network Authentication Service
   Administration for information about how z/OS Network Authentication
   Service uses keys and how to customize environment variables related to
   keys.

KERBNAME(kerberos-realm-name)
   Specifies the local realm name or a trust relationship for z/OS Network
   Authentication Service. The maximum length of this field is 117 characters.

   • When you specify the local realm name for the KERBDFLT realm, you
     must specify KERBNAME using the unqualified form of the local realm
     name. For example:

     RDEFINE REALM KERBDFLT KERB(KERBNAME(KRB2000.IBM.COM)
When you specify a trust relationship, you must specify the fully qualified principal name using the following form:

/.../kerberos_realm_name_1/krbtgt/kerberos_realm_name_2

For more information about defining trust relationships, see z/OS Integrated Security Services Network Authentication Service Administration.

Syntax rules for naming your local realm:

The local realm name that you define to RACF can consist of any character, except the / (X'61') character. You can enter the name with or without single quotation marks, depending on the following:

- If parentheses, commas, blanks, or semicolons are entered as part of the name, the character string must be enclosed in single quotation marks.
- If a single quotation mark is intended to be part of the name and the entire character string is enclosed in single quotation marks, you must use two single quotation marks together to represent each single quotation mark within the string.
- If the first character of the name is a single quotation mark, you must enter the string within single quotation marks, with two single quotation marks entered for the single quotation mark.

Regardless of the case in which it is entered, RACF translates the name of the local z/OS Network Authentication Service realm to upper case. However, RACF does not ensure that a valid kerberos-realm-name has been specified.

Guidelines for naming your local realm:

- Avoid using EBCDIC variant characters to prevent problems with different code pages.
- Carefully consider the length of the local realm name. Its length limits the length of local principal names because fully qualified local principal names use the following form and cannot exceed 240 characters:

/.../kerberos_realm_name/principal_name

The length of the fully qualified local principal name is checked by RACF only when a local kerberos-principal-name is added or altered. Therefore, plan ahead to ensure that the maximum length of your principal names is sufficient and help you avoid renaming the local realm. If you rename your local realm (using the RALTER command), the keys for existing principals become unusable.

**MAXTKTLFE**(max-ticket-life)

Specifies the max-ticket-life for the local z/OS Network Authentication Service in seconds. The value of MAXTKTLFE is 1–2 147 483 647. Note that 0 is not a valid value.

This keyword is only applicable when defining the KERBDFLT REALM profile for the local z/OS Network Authentication Service realm.

If MAXTKTLFE is specified, DEFTKTLFE and MINTKTLFE must also be specified.

**MINTKTLFE**(min-ticket-life)

Specifies the min-ticket-life for the z/OS Network Authentication Service in seconds. The value of MINTKTLFE is 1–2 147 483 647. Note that 0 is not a valid value.
This keyword is only applicable when defining the KERBDFLT REALM profile for the local Kerberos realm.

If MINTKTLFE is specified, DEFTKTLFE and MAXTKTLFE must also be specified.

**PASSWORD( kerberos-password )**
Specifies the value of the `kerberos-password`. The maximum length of this value is 128 characters. The PASSWORD keyword is applicable to all REALM class profile definitions. A password must be associated with the definition of a trust relationship or else the definition is incomplete.

**Guideline:** Avoid using EBCDIC variant characters to prevent problems with different code pages.

The password that you define to RACF can consist of any character. You can enter a password with or without single quotation marks, depending on the following:

- If parentheses, commas, blanks, or semicolons are entered as part of the password, the character string must be enclosed in single quotation marks.
- If a single quotation mark is intended to be part of the password and the entire character string is enclosed in single quotation marks, you must use two single quotation marks together for each single quotation mark within the string.
- If the first character of the password is a single quotation mark, you must enter the string within single quotation marks, with two single quotation marks entered for the character.

Both uppercase and lowercase characters are accepted and maintained in the case in which they are entered.

**Note:** This keyword is intended for administrators to be able to associate a `kerberos-password` with the definition of a realm. It is not the same as a RACF user password and is not constrained by the SETROPTS password rules or change interval values that might be established for RACF user passwords.

**LEVEL( nn )**
Specifies a level indicator, where `nn` is an integer from 0–99. The default is 0.

Your installation assigns the meaning of the value. It is included on all records that log resource accesses and is listed by the RLIST command.

**NOTIFY([userid])**
Specifies the user ID of a user to be notified whenever RACF uses this profile to deny access to a resource. If you specify NOTIFY without specifying a user ID, RACF takes your user ID as the default; you are notified whenever the profile denies access to a resource.

A user who is to receive NOTIFY messages should log on frequently to take action in response to the unauthorized access attempt described in each message. RACF sends NOTIFY messages to the SYS1.BROADCAST data set. When the resource profile also includes WARNING, RACF might have granted access to the resource to the user identified in the message.

When RACF denies access to a resource, it does *not* notify a user:

- When the resource is in the PROGRAM class
- When the resource is in a class for which an application has built in-storage profiles using RACROUTE REQUEST=LIST
RDEFINE

Some applications, such as IMS and CICS, load all the profiles for a given class into storage. After these profiles are in storage, the applications can do a fast authorization check using RACROUTE REQUEST=FASTAUTH. One difference is that, in some cases, fast authorization checking does not issue warning messages, notification messages, or support auditing. In cases where it does not, return and reason codes are returned to the application to allow support of these functions. Return and reason codes are returned to the application to allow support of these functions. The application can examine the return and reason codes and use RACROUTE REQUEST=AUTH to create the messages and audit records. If the application uses RACROUTE REQUEST=AUTH to support auditing, the specified user is notified. Otherwise, notification, warning, and such does not occur.

For details on using RACF with IMS, see z/OS Security Server RACF Security Administrator’s Guide. For details on using RACF with CICS, refer CICS RACF Security Guide.

• When the profile is used to disallow the creation or deletion of a data set, NOTIFY is used only for resource access checking, not for resource creation or deletion.

OWNER(userid or group-name)

Specifies a RACF-defined user or group to be assigned as the owner of the resource you are defining. If you omit this operand, you are defined as the owner. The user specified as the owner does not automatically have access to the resource. Use the PERMIT command to add the owner to the access list as desired.

PROXY

Specifies information which the z/OS LDAP server will use when acting as a proxy on behalf of a requester. The R_proxy.serv (IRRSPY00) SAF callable service will attempt to retrieve this information when it is not explicitly supplied with the invocation parameters. Applications or other services which use the R_proxy.serv callable service, such as IBM Policy Director Authorization Services for z/OS and OS/390, may instruct their invokers to define PROXY segment information.

LDAPHOST(ldap_url)

Specifies the URL of the LDAP server which the z/OS LDAP server will contact when acting as a proxy on behalf of a requester. An LDAP URL has a format such as ldap://123.45.6:389 or ldaps://123.45.6:636, where ldaps indicates that an SSL connection is desired for a higher level of security. LDAP will also allow you to specify the host name portion of the URL using either the text form (BIGHOST.POK.IBM.COM) or the dotted decimal address (123.45.6). The port number is appended to the host name, separated by a colon (X’7A’).

For more information about LDAP URLs and how to enable LDAP servers for SSL connections, see IBM Tivoli Directory Server Administration and Use for z/OS or z/OS Integrated Security Services LDAP Server Administration and Use.

The LDAP URL that you define to RACF can consist of 10–1023 characters. A valid URL must start with either ldap:// or ldaps://. RACF will allow any characters to be entered for the remaining portion of the URL, but you should ensure that the URL conforms to TCP/IP conventions. For example, parentheses, commas, blanks, semicolons, and single quotation marks should be escaped.
marks are not typically allowed in a host name. The LDAP URL can be entered with or without single quotation marks, however, in both cases, it will be translated to uppercase.

RACF does not ensure that a valid LDAP URL has been specified.

**BINDDN** *(bind_distinguished_name)*

Specifies the distinguished name (DN) which the z/OS LDAP server will use when acting as a proxy on behalf of a requester. This DN will be used in conjunction with the BIND password, if the z/OS LDAP server needs to supply an administrator or user identity to BIND with another LDAP server. A DN is made up of attribute value pairs, separated by commas. For example:

```
cn=Ben Gray,ou=editing,o=New York Times,c=US
cn=Lucille White,ou=editing,o=New York Times,c=US
cn=Tom Brown,ou=reporting,o=New York Times,c=US
```

When you define a BIND DN to RACF, it can contain 1–1023 characters. The BIND DN can consist of any characters and can be entered with or without single quotation marks. The following rules apply:

- If parentheses, commas, blanks, or semicolons are to be entered as part of the BIND DN, the character string must be enclosed in single quotation marks.
- If a single quotation mark is intended to be part of the BIND DN, use two single quotation marks together for each single quotation mark within the string, and enclose the entire string within single quotation marks.

Both uppercase and lowercase characters are accepted and maintained in the case in which they are entered. For more information about LDAP distinguished names, see **IBM Tivoli Directory Server Administration and Use for z/OS** or **z/OS Integrated Security Services LDAP Server Administration and Use**.

If you issue the RDEFINE command as a RACF operator command and you specify the BIND DN in lowercase, you must include the BIND DN within single quotations.

RACF does not ensure that a valid BIND DN has been specified.

**BINDPW**

Specifies the password which the z/OS LDAP server will use when acting as a proxy on behalf of a requester.

When you define a BIND password to RACF, it can contain 1–128 characters. The BIND password can consist of any characters (see exception below) and can be entered with or without single quotation marks. The following rules apply:

- The BIND password cannot start with a left brace `{` character (X'8B').
- If parentheses, commas, blanks, or semicolons are to be entered as part of the BIND password, the character string must be enclosed in single quotation marks.
- If a single quotation mark is intended to be part of the BIND password, use two single quotation marks together for each single quotation mark within the string, and enclose the entire string within single quotation marks.

Both uppercase and lowercase characters are accepted and maintained in the case in which they are entered. For more information about LDAP
If you issue the RDEFINE command as a RACF operator command and you specify the BIND password in lowercase, you must include the BIND password within single quotations.

RACF does not ensure that a valid BIND password has been specified.

Attention:

- When the command is issued from ISPF, the TSO command buffer (including possible BINDPW password data) is written to the ISPLOG data set. As a result, you should not issue this command from ISPF or you must control the ISPLOG data set carefully.
- When the command is issued as a RACF operator command, the command and the possible BINDPW password data is written to the system log. Therefore, use of RDEFINE as a RACF operator command should either be controlled or you should issue the command as a TSO command.

SECLABEL(seclabel-name)

Specifies the installation-defined security label for this profile.

A security label corresponds to a particular security level (such as CONFIDENTIAL) with a set of zero or more security categories (such as PAYROLL or PERSONNEL).

RACF stores the name of the security label you specify in the resource profile if you are authorized to use that SECLABEL.

If you are not authorized to the SECLABEL or if the name you had specified is not defined as a SECLABEL profile in the SECLABEL class, the resource profile is not created.

If SECLABEL is not specified, the created profile will not have a SECLABEL associated with the resource, unless the SETROPTS MLACTIVE option is turned on. In this case, the user's current logon SECLABEL will automatically be assigned to the profile.

SECLEVEL(seclevel-name)

Specifies the name of an installation-defined security level. The name corresponds to the number that is the minimum security level that a user must have to access the resource. The seclevel-name must be a member of the SECLEVEL profile in the SECDATA class.

When you specify SECLEVEL and the SECDATA class is active, RACF adds security level checking to its other authorization checking. If global access checking grants access, RACF compares the security level allowed in the user profile with the security level required in the resource profile. If the security level in the user profile is less than the security level in the resource profile, RACF denies the access. If the security level in the user profile is equal to or greater than the security level in the resource profile, RACF continues with other authorization checking. The SECLEVEL operand is required for the SECLABEL class.

Note: RACF does not perform security level checking for a started task that has the RACF privileged or trusted attribute. The RACF privileged or trusted attribute can be assigned to a started task through the RACF
RDEFINE

started procedures table or STARTED class. Also, RACF does not
enforce security level information specified on profiles in the PROGRAM
class.

If the SECDATA class is not active, RACF stores the name you specify in the
resource profile. When the SECDATA class is activated and the name you
specified is defined as a SECLEVEL profile, RACF can perform security level
access checking for the resource profile. If the name you specify is not defined
as a SECLEVEL profile, you are prompted to provide a valid SECLEVEL name.

SESSION
Is only valid for the APPCLU resource class. It specifies that when changing an
APPCLU class profile, the following suboperands add, change, or delete
SESSION segment field values. The SESSION segment is used to control the
establishment of sessions between logical units under LU6.2.

CONVSEC
Specifies the level or levels of security checking performed when
conversations are established with the LU protected by this profile.

Guideline: Specify a CONVSEC option for each APPCLU profile.

ALREADYV
APPC/MVS RACF does not verify the user ID and password for any
inbound allocate requests. If you specify ALREADYV, you assume that
user IDs and passwords have already been verified by the partner LU.
You must specify this only if the partner LU is trustworthy.

AVPV
The user ID/password is already verified and persistent verification is
requested.

CONV
APPC/MVS issues a RACROUTE REQUEST=VERIFY to verify the user
ID and password for all inbound allocate requests.

NONE
All inbound allocate requests pass without RACF checking for a valid
user ID. No RACROUTE REQUEST=VERIFY is issued.

PERSISTV
Specifies persistent verification.

INTERVAL(n)
Sets the maximum number of days the session key is valid. The n value
can be 1–32767. If the key interval is longer than the installation maximum
(set with SETROPTS SESSIONINTERVAL), then the profile is created.

If the key interval is not specified and there is a SETROPTS
SESSIONINTERVAL value, the profile is created with that value. If there is
no SETROPTS SESSIONINTERVAL value, there is no limit to the number
of days the session key is valid.

LOCK
Mark the profile as locked. This prevents all session establishment from
succeeding.

SESSKEY(session-key)
Change the key for this profile. The session-key value can be expressed in
two ways:
• X’y’ where y is a hexadecimal number of 1–16 digits
• z or ’z’ where z is a string of 1–8 characters
If the entire 16 digits or 8 characters are not used, the field is padded to the right with binary zeros.

**Note:** Session keys are 64-bit Data Encryption Standard (DES) keys. With DES, 8 of the 64 bits are reserved for use as parity bits, so those 8 bits are not part of the 56-bit key. In hexadecimal notation, the DES parity bits are: X'0101 0101 0101 0101'. Any two 64-bit keys are equivalent DES keys if their only difference is in one or more of these parity bits. For instance, the following SESSKEY values, although appearing to be quite different, are equivalent because they differ only in the last bit of each byte:

- BDF0KMD4, which is X'C2C4 C6F0 D2D4 F4D8'
- CEG1LN5R, which is X'C3C5 C7F1 D3D5 F5D9'

**SIGVER**

Specifies the options for verifying the signatures of programs that are protected by this general resource profile.

**Rule:** Specify SIGVER only for profiles in the PROGRAM class. Any options specified with the SIGVER operand are ignored for profiles in a class other than the PROGRAM class.

**Restriction:** Digital signature verification is supported only for program objects that are stored as members of a partitioned data set extended (PDSE) library. Digital signature verification is *not* supported for programs that are stored as members of a partitioned data set (PDS) library.

Any options specified with the SIGVER operand are ignored for unsupported programs.

**Note:** Regardless of the SIGREQUIRED setting, specifying FAILLOAD(NEVER) and SIGAUDIT(NONE) is equivalent to having no SIGVER segment.

For detailed information, see "Program signing and verification" in [z/OS Security Server RACF Security Administrator’s Guide](https://www.ibm.com/support/knowledgecenter/SSEPGG_11.1.0/security/serverracf manuals.html).

**SIGREQUIRED**

Specifies whether programs that are protected by this profile must be digitally signed.

**YES**

Specifies that programs must be digitally signed.

When you specify SIGREQUIRED(YES), the following conditions apply to any program that is protected by this general resource profile:

- If the program has a digital signature:
  - Signature verification processing occurs.
  - The program continues to load according to the FAILLOAD setting.
  - Logging occurs according to the SIGAUDIT setting.
- If the program has no digital signature:
  - Signature verification processing occurs, resulting in a signature verification failure.
  - The program continues to load according to the FAILLOAD setting.
  - Logging occurs according to the SIGAUDIT setting.

**NO**

Specifies that programs need not be digitally signed.
When you specify SIGREQUIRED(NO), the following conditions apply to any program that is protected by this general resource profile:

- If the program has a digital signature:
  - Signature verification processing occurs.
  - The program continues to load according to the FAILLOAD setting.
  - Logging occurs according to the SIGAUDIT options.
- If the program has no digital signature:
  - No signature verification occurs.
  - The program continues to load. The FAILLOAD setting is ignored.
  - No logging occurs. The SIGAUDIT setting is ignored.

If SIGREQUIRED is not specified, SIGREQUIRED(NO) is the default value.

**FAILLOAD**
Specifies the conditions under which the program fails to load in the event that a signature verification failure occurs.

**ANYBAD**
Specifies that the program fails to load when a signature verification failure occurs, regardless of the cause. Such failures include those resulting from an incorrect signature, or an error establishing the trust of the signer. This setting includes failures related to administrative errors, such as a missing or incorrectly defined key ring.

The ANYBAD setting includes the failures covered by the BADSIGONLY setting, and also includes errors establishing the trust of the signer.

**BADSIGONLY**
Specifies that the program fails to load only when the signature verification failure is caused by an incorrect digital signature. Such failures include only those resulting from a signature that fails verification or a signature structure that is missing or improperly formatted.

In contrast to ANYBAD, the BADSIGONLY setting does not cause a program to fail to load when the program has a valid signature originating from an untrusted signer.

**NEVER**
Specifies that the program never fails to load when a signature verification failure is detected.

If FAILLOAD is not specified, FAILLOAD(NEVER) is the default value.

**SIGAUDIT**
Specifies which signature verification events are logged. Messages are issued to the console only for signature verification failures that are logged.

**ALL**
Logs all signature verifications, whether successful or not.

**SUCCESS**
Logs only signature verification successes. In other words, the digital signature is valid and the root CA certificate is trusted.

**ANYBAD**
Logs all signature verification failures, regardless of the cause of the failure. Such failures include those resulting from an incorrect signature,
or an error establishing the trust of the signer. This setting includes failures related to administrative errors, such as a missing or incorrectly defined key ring.

The ANYBAD setting logs the failures covered by the BADSIGONLY setting, and also logs errors encountered when establishing the trust of the signer.

**BADSIGONLY**

Logs only signature verification failures caused by an incorrect digital signature. Such failures include only those resulting from a signature that fails verification or a signature structure that is missing or improperly formatted.

In contrast to ANYBAD, the BADSIGONLY setting does not log a signature verification failure when the program has a valid signature originating from an untrusted signer.

**NONE**

Logs no digital signature verification events.

If SIGAUDIT is not specified, SIGAUDIT(NONE) is the default value.

**SINGLEDSN**

Specifies that the tape volume can contain only one data set. SINGLEDSN is valid only for a TAPEVOL profile. If the volume already contains more than one data set, RACF issues a message and ignores the operand.

**SSIGNON**

Defines the application key or a secured signon key and indicates the method you want to use to protect the key value within the RACF database on the host. When defining the profile, you can either mask or encrypt the key. The *key-value* represents a 64-bit (8-byte) key that must be represented as 16 hexadecimal characters. The valid characters are 0–9 and A–F.

**Restrictions:**

- As with RACF passwords, the database unload facility does not unload application keys or the secured signon keys.
- The RLIST command does not list the value of the application keys or a secured signon keys. Therefore, when you define the keys, you should note the value and keep it in a secure place.

*(key-value)*

Specifies that you want to mask the key value using the masking algorithm.

**Rules:**

- You can specify this operand only once for each application key.
- If you mask a key, you *cannot* encrypt it. These are mutually exclusive.

You can use the RLIST command to verify that the key is protected.

**KEYENCRYPTED(key-value)**

Specifies that you want to encrypt the key value.

**Rules:**

- You can specify this operand only once for each application key.
- If you encrypt a key, you *cannot* mask it. These are mutually exclusive.
- A cryptographic product must be installed and active on the system.

You can use the RLIST command to verify that the key is protected.
STDATA

Used to control security for started tasks. STDATA should only be specified for profiles in the STARTED class.

USER

USER(userid)

Specifies the user ID to be associated with this entry.

RACF issues a warning message if the specified userid does not exist, or if the USER operand is not specified, but data is placed into the STDATA segment. If the error is not corrected, RACF uses the started procedures table to process START requests that would have used this STARTED profile.

USER(=MEMBER)

Specifies that the procedure name should be used as the user ID. If =MEMBER is specified for USER, a group-name value should be specified for the GROUP operand. If =MEMBER is specified for both USER and GROUP, a warning message is issued and problems might result when the profile is used. For information, see z/OS Security Server RACF System Programmer's Guide.

GROUP

GROUP(group-name)

Specifies the group name to be associated with this entry.

RACF issues a warning message if the specified group-name does not exist. If userid and group-name are specified, RACF verifies that the user is connected to the group. If GROUP is specified incorrectly, the started task runs as an undefined user.

GROUP(=MEMBER)

Specifies that the procedure name should be used as the group name. If =MEMBER is specified for GROUP, a userid value must be specified for the USER operand or RACF uses the started procedures table to assign an identity for this started task. If =MEMBER is specified for both USER and GROUP, a warning message is issued and problems might result when the profile is used. For information, see z/OS Security Server RACF System Programmer's Guide.

If GROUP is not specified the started task runs with the default group of the specified user ID.

PRIVILEGED( YES | NO )

Specifies whether the started task should run with the RACF PRIVILEGED attribute. The PRIVILEGED attribute allows the started task to pass most authorization checking. No installation exits are called, no SMF records are generated, and no statistics are updated. (Note that bypassing authorization checking includes bypassing the checks for security classification of users and data.) For more information, see z/OS Security Server RACF System Programmer's Guide.

If PRIVILEGED(NO) is specified, the started tasks runs without the RACF PRIVILEGED attribute.

If PRIVILEGED is not specified PRIVILEGED(NO) is defaulted.

TRACE( YES | NO )

Specifies whether a message should be issued to the operator when this entry is used to assign an ID to the started task.
If TRACE(YES) is specified, RACF issues an informational message to the operator to record the use of this entry when it is used to assign an ID to a started task. This record can be useful in finding started tasks that do not have a specific entry defined and in diagnosing problems with the user IDs assigned for started tasks.

If TRACE(NO) is specified, RACF does not issue an informational message when this entry is used.

If TRACE is not specified, TRACE(NO) is defaulted.

TRUSTED(YES | NO)
Specifies whether the started task should run with the RACF TRUSTED attribute. The TRUSTED attribute is similar to the PRIVILEGED attribute except that auditing can be requested using the SETROPTS LOGOPTIONS command. For more information about the TRUSTED attribute, see z/OS Security Server RACF System Programmer’s Guide.

If TRUSTED(NO) is specified, the started tasks runs without the RACF TRUSTED attribute.

If TRUSTED is not specified, TRUSTED(NO) is defaulted.

SVFMR
Defines profiles associated with a particular SystemView for MVS application.

SCRIPTNAME(script-name)
Specifies the name of the list of default logon scripts associated with this application. This operand is optional. If this operand is omitted, no scripts are associated with the application.

The script-name is a 1–8 character alphanumeric name of a member of an MVS partitioned data set (PDS). RACF accepts both uppercase and lowercase characters for script-name, but lowercase characters are translated to uppercase.

The PDS member specified by the script-name contains a list of other PDS members that contain the scripts associated with this application’s profile. The PDS and members, including the member that contains the list of other members, are created by the SystemView for MVS administrator.

PARMNAME(parm-name)
Specifies the name of the parameter list associated with this application. If this operand is omitted, no parameters are associated with the application.

The parm-name is a 1–8 character alphanumeric name of a member of an MVS partitioned data set (PDS). RACF accepts both uppercase and lowercase characters for parm-name, but lowercase characters are translated to uppercase.

The PDS member specified by parm-name contains a list of other PDS members that contain the parameters associated with this application’s profile. The PDS and members, including the list of other members, are created by System View for the MVS administrator.

TIMEZONE({E | W} hh[,mm])
Specifies the time zone in which a terminal resides. TIMEZONE is valid only for resources in the TERMINAL class; RACF ignores it for all other resources.

Specify TIMEZONE only when the terminal is not in the same time zone as the processor on which RACF is running and you are also specifying WHEN to limit access to the terminal to specific time periods. In this situation, TIMEZONE provides the information RACF needs to calculate the time values correctly. If
you identify more than one terminal in the profile-name-1 operand, all the terminals must be in the same time zone.

On TIMEZONE, you specify whether the terminal is east (E) or west (W) of the system and by how many hours (hh) and, optionally, minutes (mm) that the terminal time zone is different from the processor time zone. Valid hour values are 0–11, and valid minute values are 00–59.

For example, if the processor is in New York and the terminal is in Los Angeles, specify TIMEZONE(W 3). If the processor is in Houston and the terminal is in New York, specify TIMEZONE(E 1).

If you change the local time on the processor (to accommodate daylight savings time, for instance), RACF adjusts its time calculations accordingly. However, if the processor time zone and the terminal time zone do not change in the same way, you must adjust the terminal time zones yourself, as described earlier for the WHEN(TIME) operand.

TME
Specifies that information for the Tivoli Security Management Application be added.

Note: The TME segment fields are intended to be updated only by the Tivoli Security Management Application, which manages updates, permissions, and cross references. A security administrator should only directly update Tivoli Security Management fields on an exception basis.

All TME suboperands, with the exception of those for ROLES, can be specified when changing a resource profile in the ROLE class. Conversely, only the ROLES suboperands can be specified when changing a resource profile in any other class.

CHILDREN(profile-name ...)
Specifies the complete list of roles that inherit attributes from this role. A role is a discrete general resource profile defined in the ROLE class.

GROUPS(group-name ...)
Specifies the complete list of groups that should be permitted to resources defined in this role profile.

PARENT(profile-name)
Specifies the name of a role from which this role inherits attributes. A role is a discrete general resource profile defined in the ROLE class.

RESOURCE(resource-access-specification ...)
Specifies the complete list of resources and associated access levels for groups defined in this role profile.

One or more resource-access-specification values can be specified, each separated by blanks. Each value should contain no imbedded blanks and should have the following format:

origin-role:class-name:profile-name:authority
[:conditional-class:conditional-profile]

where origin-role is the name of the role profile from which the resource access is inherited. The class-name value is an existing resource class name and profile-name is a resource profile defined in that class. The authority is the access authority (NONE, EXECUTE, READ, UPDATE, CONTROL, or ALTER) with which groups in the role definition should be permitted to the resource.
RDEFINE

The conditional-class is a class name (APPCPORT, CONSOLE, JESINPUT, PROGRAM, TERMINAL, or SYSID) for conditional access permission, and is followed by the conditional-profile value, a resource profile defined in the conditional class.

ROLES(role-access-specification ...)
Specifies a list of roles and associated access levels related to this profile.

One or more role-access-specification values can be specified, each separated by blanks. Each value should contain no imbedded blanks and should have the following format:

role-name:authority[conditional-class:conditional-profile]

where role-name is a discrete general resource profile defined in the ROLE class. The authority is the access authority (NONE, EXECUTE, READ, UPDATE, CONTROL, or ALTER) with which groups in the role definition should be permitted to the resource.

The conditional-class is a class name (APPCPORT, CONSOLE, JESINPUT, PROGRAM, TERMINAL, or SYSID) for conditional access permission, and is followed by the conditional-profile value, a resource profile defined in the conditional class.

TVTOC
Specifies, for a TAPEVOL profile, that RACF is to create a TVTOC in the TAPEVOL profile when a user creates the first output data set on the volume.
The RDEFINE command creates a nonautomatic TAPEVOL profile; RACF creates and maintains the TVTOC for data sets residing on tape.

Specifying TVTOC also affects the access list for the TAPEVOL profile:

1. When RACF processes the RDEFINE command with the TVTOC operand, it places the user ID of the command issuer (perhaps the tape librarian) in the access list with ALTER authority.

2. When the first output data set is created on the volume, RACF adds the user ID associated with the job or task to the access list with ALTER authority.


The TVTOC operand is valid only for a discrete profile in the TAPEVOL class.

UACC(access-authority)
Specifies the universal access authority to be associated with this resource. The universal access authorities are ALTER, CONTROL, UPDATE, READ, EXECUTE (for controlled programs only), and NONE. If UACC is not specified, RACF uses the value in the ACEE or the class descriptor table. If UACC is specified without access-authority, RACF uses the value in the current connect group. For tape volumes and DASD volumes, RACF treats CONTROL authority as UPDATE authority. For all other resources listed in the class descriptor table and for applications, RACF treats CONTROL and UPDATE authority as READ authority.

If the user ID accessing the general resource has the RESTRICTED attribute, RACF treats the access authority as NONE.

WARNING
Specifies that even if access authority is insufficient, RACF is to issue a warning message and allow access to the resource. RACF also records the access attempt in the SMF record if logging is specified in the profile.
Restriction: RACF does not issue a warning message for a resource when the resource is:
- In the PROGRAM or NODES class
- In a class for which an application has built in-storage profiles using RACROUTE REQUEST=LIST.

When SETROPTS MLACTIVE(FAILURES) is in effect: A user or task can access a resource that is in WARNING mode and has no security label even when MLACTIVE(FAILURES) is in effect and the class requires security labels. The user or task receives a warning message and gains access.

Applications that use REQUEST=LIST: Some applications, such as IMS and CICS, load all the profiles for a given class into storage. After these profiles are in storage, the applications can do a fast authorization check using RACROUTE REQUEST=FASTAUTH. Fast authorization checking is different from normal authorization checking in several ways. One difference is that, in some cases, fast authorization checking does not issue warning messages, notification messages or support auditing. In cases where it does not, return and reason codes are returned to the application to allow support of these functions. The application can examine the return and reason codes and use RACROUTE REQUEST=AUTH to create the messages and audit records. If the application uses RACROUTE REQUEST=AUTH to support auditing or specifies LOG=ASIS on the RACROUTE REQUEST=FASTAUTH, the specified user is notified. Otherwise, notification, warning, and so on does not occur.

For details on using RACF with IMS, see z/OS Security Server RACF Security Administrator's Guide. For details on using RACF with CICS, refer to CICS RACF Security Guide.

WHEN
Specifies, for a resource in the TERMINAL class, the days of the week or the hours in the day when a user can access the system from the terminal. The day-of-week and time restrictions apply only when a user logs on to the system; that is, RACF does not force the user off the system if the end-time occurs while the user is logged on.

If you omit the WHEN operand, a user can access the system from the terminal at any time. If you specify the WHEN operand, you can restrict the use of the terminal to certain days of the week or to a certain time period on each day. Or, you can restrict access to both certain days of the week and to a certain time period within each day.

DAYS(day-info)
Specifies days of the week when the terminal can be used. The day-info value can be any one of the following:

ANYDAY RACF allows use of the terminal on any day. If you omit DAYS, ANYDAY is the default.

WEEKDAYS RACF allows use of the terminal only on weekdays (Monday through Friday).

day ... RACF allows use of the terminal only on the days specified, where day can be MONDAY, TUESDAY, WEDNESDAY, THURSDAY, FRIDAY, SATURDAY, or SUNDAY. You can specify the days in any order.

TIME(time-info)
Specifies the time period each day when the terminal can be used. The time-info value can be any one of the following:
ANYTIME

RACF allows use of the terminal at any time. If you omit TIME, ANYTIME is the default.

start-time:end-time

RACF allows use of the terminal only during the specified time period. The format of both start-time and end-time is hhmm, where hh is the hour in 24-hour notation (00–24) and mm is the minutes (00–59) within the range 0001–2400. Note that 2400 indicates 12:00 a.m. (midnight).

If start-time is greater than end-time, the interval spans midnight and extends into the following day.

Specifying start-time and end-time is straightforward when the processor on which RACF is running and the terminal are in the same time zone; you specify the time values in local time.

However, if the terminal is in a different time zone from the processor and you want to restrict access to certain time periods, you have two choices. You can specify the TIMEZONE operand to allow RACF to calculate the time and day values correctly. Otherwise, you can adjust the time values yourself, by translating the start-time and end-time for the terminal to the equivalent local time for the processor.

For example, assume that the processor is in New York and the terminal is in Los Angeles, and you want to allow access to the terminal from 8:00 A.M. to 5:00 P.M. in Los Angeles. In this situation, you would specify TIME(1100:2000). If the processor is in Houston and the terminal is in New York, you would specify TIME(0900:1800).

If you omit DAYS and specify TIME, the time restriction applies to all seven days of the week. If you specify both DAYS and TIME, RACF allows use of the terminal only during the specified time period and only on the specified days.

Examples

Table 85. RDEFINE examples

Example 1

Operation User TBK20 wants to define resource GIMS600 in class GIMS which is a resource group class. He also wants to define TIMS200, TIMS111, TIMS300, and TIMS333 as members of the resource group (GIMS600).

Known User TBK20 has the CLAUTH attribute for the GIMS and TIMS classes. GIMS is a resource group class, and TIMS is its associated resource member class. TIMS200 and TIMS111 are members of another resource group. The user has ALTER authority to the other resource group. User TBK20 wants to issue the command as a RACF TSO command.

Command RDEFINE GIMS GIMS600 ADDMEM(TIMS200 TIMS111 TIMS300 TIMS333)

Defaults OWNER (TBK20) LEVEL(0) AUDIT(FAILURES(READ)) UACC(NONE)

Example 2

Operation User ADM1 wants to define a generic profile for all resources starting with a T belonging to the TIMS class, and to require that users must reenter their passwords whenever they enter any IMS transaction starting with a T.

Known User ADM1 has the SPECIAL attribute. User ADM1 wants to issue the command as a RACF TSO command.

Command RDEFINE TIMS T* APPL('REVERIFY')

Defaults UACC(NONE) APPL(ADM1) LEVEL(0) AUDIT(FAILURES(READ))
Table 85. RDEFINE examples (continued)

Example 3

**Operation**
User ADM1 wants to define AMASPZAP as a controlled program with program-accessed data set checking.

**Known**
User ADM1 has the SPECIAL attribute. AMASPZAP resides in SYS1.LINKLIB on the SYSRES volume. RACF program control is active. User ADM1 wants to issue the command as a RACF TSO command.

**Command**
RDEFINE PROGRAM AMASPZAP ADDMEM('SYS1.LINKLIB'/SYSRES/PADCHK)

**Defaults**
UACC(NONE) OWNER(ADM1) LEVEL(0) AUDIT(FAILURES(READ))

Example 4

**Operation**
User ADM1 wants to define all load modules that start with IKF as controlled programs that do not require program-accessed data set checking.

**Known**
User ADM1 has the SPECIAL attribute. All load modules whose names begin with IKF reside in SYS1.COBLIB on the SYSRES volume. User ADM1 wants to issue the command as a RACF operator command, and the RACF subsystem prefix is @.

**Command**
@RDEFINE PROGRAM IKF* ADDMEM('SYS1.COBLIB'/SYSRES/NOPADCHK)

**Defaults**
UACC(NONE) OWNER(ADM1) LEVEL(0) AUDIT(FAILURES(READ))

Example 5

**Operation**
User JPQ12 wants to define a tape volume labeled DP0123 and allow it to hold a TVTOC. The tape volume is assigned a UACC of NONE.

**Known**
User JPQ12 has the SPECIAL attribute. User JPQ12 wants to issue the command as a RACF TSO command.

**Command**
RDEFINE TAPEVOL DP0123 TVTOC UACC(NONE)

**Defaults**
OWNER(JPQ12) LEVEL(0) AUDIT(FAILURES(READ))

Example 6

**Operation**
User ADM1 wants to prepare the TCICSTRN class to be used for RACGLIST processing.

**Known**
User ADM1 has the SPECIAL attribute. User ADM1 wants to issue the command as a RACF TSO command.

**Command**
RDEFINE RACGLIST TCICSTRN UACC(NONE)

**Defaults**
OWNER(ADM1) LEVEL(0) AUDIT(FAILURES(READ))

Example 7

**Operation**
The security administrator wants to define a profile for TSO in the PTKTDATA class. The security administrator wants to direct the command to run under the authority of user OJC11 at node NYTSO.

**Known**
ELVIS1 is the user ID of the security administrator.

OJC11 has the SPECIAL attribute on node NYTSO.

The profile name is TSOR001.

The key-value is e001193519561977 and is to be masked. The security administrator wants to issue the command as a RACF TSO command.

The security administrator and OJC11 at NYTSO have an already established user ID association.

**Command**
RDEFINE PTKTDATA TSOR001 SSIGNON(KEYMASKED(e001193519561977)) AT(NYTSO,OJC11)

**Defaults**
UACC(NONE)

Example 8

**Operation**
The security administrator wants to create an entry in the dynamic started procedures table for the OMVS started procedure by defining a generic profile in the STARTED class.

**Known**
The administrator wants to use the procedure name as the user ID. The group name is STCGRP.

SETROPTS GENERIC(STARTED) has been issued to allow generic profiles to be created in this class. The security administrator wants to issue the command as a RACF TSO command.

**Command**
RDEFINE STARTED OMVS.* STDATA(USER(=MEMBER) GROUP(STCGRP))

**Defaults**
PRIVILEGED(NO) TRACE(NO) TRUSTED(NO) UACC(NONE)
Table 85. RDEFINE examples (continued)

Example 9  
Operation User ADM1 wants to define the following:
- A SystemView for the MVS application named APPL1.HOST1.USER1
- TSOR220 application data
- A list of scripts named APPL1SC for the application
- A list of parameters named APPL1P for the application

Known User ADM1 has CLAUTH authority for the SYSMVIEW class.

Command RDEFINE SYSMVIEW APPL1.HOST1.USER1 APPLDATA('TSOR220')
  SVFMR(SCRIPTNAME(APPL1SC) PARMNAME(APPL1P))

Defaults UACC(NONE)

Example 10  
Operation Local realm krb2000.ibm.com is being defined with a minimum ticket lifetime of 5 minutes, a default ticket lifetime of 10 hours, a maximum ticket lifetime of 24 hours, and a password of 744275. All of the ticket lifetime values are specified in seconds.

Known The administrator has access to the KERBDFLT profile in the REALM class.

Command RDEFINE REALM KERBDFLT KERB(KERBNAME(krb2000.ibm.com)
  MINTKTLFE(300) DEFTKTLFE(36000)
  MAXTKTLFE(86400) PASSWORD(744275))

Defaults None.

Example 11  

Known The administrator has access to the /.../KERB390.ENDICOTT.IBM.COM/KERBTGT/
  KER2000.ENDICOTT.IBM.COM profile in the REALM class.

Command RDEFINE REALM /.../KERB390.ENDICOTT.IBM.COM/KERBTGT/
  KER2000.ENDICOTT.IBM.COM KERB(PASSWORD(12345678))

Defaults None.

Example 12  
Operation The administrator wants to create a profile (TSOIM13) in the PTKTDATA class with replay protection bypassed.

Known The administrator has the SPECIAL attribute.

Command RDEFINE PTKTDATA TSOIM13 APPLDATA('NO REPLAY PROTECTION')

Defaults None.

Example 13  
Operation The administrator is defining the system wide defaults for Enterprise Identity Mapping (EIM) applications. One of the applications uses the default name given to RACF.

Known The EIM domain’s distinguished name is ibm-eimDomainName=Pok EIM Domain,o=IBM,c=US. The domain resides in LDAP at http://some.big.host/. The bind distinguished name has authority to retrieve lookup information. The name given to the local RACF registry is RACFSYS2.

Command RDEFINE FACILITY IRR.PROXY.DEFualts EIM(
  DOMAINDN('ibm-eimDomainName=Pok EIM Domain,o=IBM,c=US')
  OPTIONS(ENABLE)
  LOCALREGISTRY(RACFSYS2))

Defaults None.

Example 14  
Operation The administrator wants to define the SAFDFLT profile in the REALM class using the APPLDATA field to define the RACF realm name.

Known The administrator has the SPECIAL attribute. The realm name racf.winmvs2c is selected by the security administrator to give a name to the set of user ids and other user information held in the security manager database. If Kerberos is in use in the installation, the Kerberos realm name would be expected to be different than the SAFDFLT realm name.

Command RDEFINE REALM SAFDFLT APPLDATA('racf.winmvs2c')

Defaults None.
### Example 15
**Operation**
The security administrator with user ID ADMIN1 wants to add a new class to the class descriptor table (CDT) named TSTCLAS8.

**Known**
The administrator has the SPECIAL attribute.

**Command**
RDEFINE CDT TSTCLAS8 UACC(NONE)
CDTINFO(DEFAULTUACC(NONE) FIRST(ALPHA) MAXLENGTH(42)
OTHER(ALPHA,NUMERIC,SPECIAL) POSIT(303) RAclist(REQUIRED)
SECLABELSREQUIRED(YES))

**Important:** The dynamic CDT must be built or refreshed to make this change effective. Use the SETROPTS RACLIST(CDT) or the SETROPTS RACLIST(CDT) REFRESH command.

**Defaults**
AUDIT(FAILURES(READ)) OWNER(ADMIN1) LEVEL(0) CDTINFO(CASE(UPPER)
DEFAULTRC(4) GENLIST(DISALLOWED) KEYQUALIFIERS(0) MACPROCESSING(NORMAL)
OPERATIONS(NO) PROFILESALLOWED(YES) SIGNAL(NO))

### Example 16
**Operation**
The security administrator Rui wants to specify that the identity cache should never store a mapping to a local z/OS user ID when it is provided by an application. The identity cache must always use EIM to find a mapping and it must always reject a store request if it cannot find a mapping.

**Known**
At Rui's installation, identity mappings in EIM are not changed frequently so the default MAPPINGTIMEOUT value of 3600 seconds (one hour) is acceptable.

**Command**
RDEFINE LDAPBIND IRR.ICTX.DEFAULTS
ICTX(USEMAP(NO) DMAP(YES) MAPREQUIRED(YES))

**Defaults**
MAPPINGTIMEOUT defaults to 3600 seconds.

### Example 17
**Operation**
Rui wants to protect a DB2 table owned by ZHAOHUI by defining a general resource called DSN.ZHAOHUI.TABLE.ALTER in the MDSNTB class.

**Known**
Rui's user ID ADMRUI has the SPECIAL attribute. The installation uses the DB2 RACF access control module (ACM) and the ACM is configured for multiple-subsystem scope.

**Command**
RDEFINE MDSNTB DSN.ZHAOHUI.TABLE.ALTER UACC(NONE)

**Defaults**
OWNER(ADMRUI) LEVEL(0) AUDIT(FAILURES(READ))

### Example 18
**Operation**
User SECADM wants to define a custom field to store employee home addresses in the CSDATA segment of her user profiles. The custom field will be named ADDRESS. It will be a character field and will contain a quoted string.

**Known**
The user has the SPECIAL attribute. The new custom field is not effective until the system programmer rebuilds the dynamic parse table using the IRRDPI00 UPDATE command.

**Command**
RDEFINE CFIELD USER.CSDATA.ADDRESS UACC(NONE)
CFDEF(TYPE(CHAR)
MAXLENGTH(100)
FIRST(ANY)
OTHER(ANY)
HELP('EMPLOYEE''S HOME ADDRESS. SPECIFY UP TO 100 CHARACTERS.'
MIXED(YES)
LISTHEAD('HOME ADDRESS ='))

**Defaults**
AUDIT(FAILURES(READ)) OWNER(SECADM) LEVEL(0)

### Example 19
**Operation**
User SECADM wants to control the XYZLIB64 program and specify that it must be digitally signed before it can be loaded, that the program should fail to load if its digital signature cannot be verified for any reason, and that logging of signature verification events should occur for only failures. The XYZLIB64 program does not require program-accessed data set checking.

**Known**
The user has the SPECIAL attribute. The XYZLIB64 program is a program object that resides in a partitioned data set extended (PDSE) library named SYS1.XYZ.LOADDLL.

**Command**
RDEFINE PROGRAM XYZLIB64 UACC(READ)
ADDMEM('SYS1.XYZ.LOADDLL'//NOPADCHK)
SIGHDR(SIGREQUIRED(YES) FAILLOAD(ANYBAD) SIGAUDIT(ANYBAD))

**Defaults**
AUDIT(FAILURES(READ)) OWNER(SECADM) LEVEL(0)
RDELETE

RDELETE (Delete general resource profile)

Purpose

Use the RDELETE command to delete RACF resources belonging to classes specified in the class descriptor table.

This command removes the profile for the resource from the RACF database.

To have changes take effect after deleting a generic profile, if the class is not RACLISTed by either the SETROPTS RACLST or RACROUTE REQUEST=LIST,GLOBAL=Yes, one of the following steps is required:

- The security administrator issues the SETROPTS command:

  SETROPTS GENERIC(class-name) REFRESH

  See the SETROPTS command for authorization requirements.

- The user of the resource logs off and logs on again.

To have changes take effect after deleting a generic profile if the class is RACLISTed, the security administrator issues the following command:

SETROPTS RACLST(class-name) REFRESH

For more information, refer to z/OS Security Server RACF Security Administrator’s Guide.

Issuing options

The following table identifies the eligible options for issuing the RDELETE command:

<table>
<thead>
<tr>
<th>As a RACF TSO command?</th>
<th>As a RACF operator command?</th>
<th>With command direction?</th>
<th>With automatic command direction?</th>
<th>From the RACF parameter library?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

For information on issuing this command as a RACF TSO command, refer to Chapter 3, “RACF TSO commands,” on page 15.

For information on issuing this command as a RACF operator command, refer to Chapter 4, “RACF operator commands,” on page 21.

You must be logged on to the console to issue this command as a RACF operator command.

Related commands

- To delete a user profile, see DELUSER (Delete user profile)
- To delete a group profile, see DELEGROUP (Delete group profile)
- To delete a data set profile, see DELDSD (Delete data set profile)
- To obtain a list of general resource profiles, see SEARCH (Search RACF database)
Authorization required

When issuing the RDELETE command as a RACF operator command, you might require sufficient authority to the proper resource in the OPERCMDS class. See \textit{z/OS Security Server RACF Security Administrator's Guide} for further information.

To remove RACF protection from a resource in a class specified in the class descriptor table, you must have sufficient authority over the resource, so that one of the following conditions is met:

- You have the SPECIAL attribute.
- The resource profile is within the scope of a group in which you have the group-SPECIAL attribute.
- You are the owner of the resource.
- If the profile is in the FILE or DIRECTRY class, the second qualifier of the profile name is your user ID.
- For a discrete profile, you are on the access list for the resource and you have ALTER authority. (If you have any other level of authority, you cannot use the command for this resource.)
- For a discrete profile, your current connect group (or, if list-of-groups checking is active, any group to which you are connected) is in the access list and has ALTER authority.
- For a discrete profile, the universal access authority for the resource is ALTER.

To specify the AT keyword, you must have READ authority to the DIRECT.node resource in the RRSFDATA class and a user ID association must be established between the specified \texttt{node.userid} pair(s).

To specify the ONLYAT keyword you must have the SPECIAL attribute, the \texttt{userid} specified on the ONLYAT keyword must have the SPECIAL attribute, and a user ID association must be established between the specified \texttt{node.userid} pair(s) if the user IDs are not identical.

Syntax

For the key to the symbols used in the command syntax diagrams, see “Syntax of RACF commands and operands” on page 9. The complete syntax of the \texttt{RDELETE} command is:

\begin{verbatim}
[subsystem-prefix]\{RDELETE | RDEL\}
  class-name
  (profile-name ...)
  [ AT([node].userid ...) | ONLYAT([node].userid ...) ]
\end{verbatim}

For information on issuing this command as a RACF TSO command, refer to Chapter 3, “RACF TSO commands,” on page 15.

For information on issuing this command as a RACF operator command, refer to Chapter 4, “RACF operator commands,” on page 21.

Parameters

\textit{subsystem-prefix}

Specifies that the RACF subsystem is the processing environment of the command. The subsystem prefix can be either the installation-defined prefix for
RDELETE

RACF (1–8 characters) or, if no prefix has been defined, the RACF subsystem name followed by a blank. If the command prefix was registered with CPF, you can use the MVS command D OPDATA to display it or you can contact your RACF security administrator.

Only specify the subsystem prefix when issuing this command as a RACF operator command. The subsystem prefix is required when issuing RACF operator commands.

**class-name**

Specifies the name of the class to which the resource belongs. Valid class names are those specified in the class descriptor table. For a list of general resource classes defined in the class descriptor table supplied by IBM, see Appendix B, “Supplied RACF resource classes,” on page 659.

**Restrictions:**

- This operand is required and must be the first operand following RDELETE.
- This command is not intended to be used for profiles in the following classes:
  - DCEUUIDS
  - DIGTCERT
  - DIGTRING
  - IDIDMAP
  - NDSLINK
  - NOTE_LINK
  - TMEADMIN
  - UNIXMAP.

**(profile-name ...)**

Specifies the name of the existing discrete or generic profile RACF is to delete from the specified class. RACF deletes the profile for any resource you name by deleting it from the RACF database. RACF uses the class descriptor table to determine if the class is defined to RACF, the syntax of resource names within the class, and whether the resource is a group.

This operand is required and must be the second operand following RDELETE.

If you specify more than one value for profile-name, you must enclose the list of names in parentheses.

Mixed-case profile names are accepted and preserved when class-name refers to a class defined in the static class descriptor table with CASE=ASIS or in the dynamic class descriptor table with CASE(ASIS).

If you specify the class-name as CACHECLS, profile-name can either be cachename_ddd_nnnnn or cachename.

Profiles in the CACHECLS hold the contents of a cache in profiles each containing 50K pieces of the cache. The profiles are named cachename_001_00001, cachename_001_00002 and so forth, for as many profiles as are needed to hold the contents of the cache, where cachename was the Cache_name given as input on the R_cacheserv callable service. RDELETE command processing for these profiles should only be used to correct an error condition, and is expected to be used in response to an IRRL100xI message that was issued in response to invocation of the R_cacheserv SAF callable service. If for some reason, you want to delete the entire cache contents (perhaps because you do not want the contents used for
authorization right after an IPL), you can delete all of the
$cachename_ddd_nnnnn$ profiles as well as the base profile by specifying just
the $cachename$ on the RDELETE.

If you specify $class-name$ as a resource grouping class, you cannot specify a
generic profile.

**Notes:**

1. If the resource you specify is a tape volume serial number that is a member
   of a tape volume set, RACF deletes the definitions for all of the volumes in
   the set.
2. RACF processes each resource you specify independently. If an error
   occurs while it is processing a resource, RACF issues a message and
   continues processing with the next resource.
3. You can use RDELETE to remove the profiles for a class defined to
   RACGLIST. For example, RDELETE RACGLIST TCICSTRN would remove the
   TCICSTRN base profile and any RACF-created TCICSTRN_nnnnn profiles
   from the RACGLIST class. If you want to stop using RACGLIST for a
   particular class, issue the command RDELETE RACGLIST $class-name$. Do
   not delete specific RACF-created profiles unless RDELETE RACGLIST
   $class-name$ was issued and failed to remove the profiles.

**AT | ONLYAT**

The AT and ONLYAT keywords are only valid when the command is issued as a
RACF TSO command.

**AT([node].userid ...)**

Specifies that the command is to be directed to the node specified by $node$,
where it runs under the authority of the user specified by $userid$ in the
RACF subsystem address space.

If $node$ is not specified, the command is directed to the local node.

**ONLYAT([node].userid ...)**

Specifies that the command is to be directed only to the node specified by
$node$ where it runs under the authority of the user specified by $userid$ in the
RACF subsystem address space.

If $node$ is not specified, the command is directed only to the local node.

**Examples**

**Table 86. RDELETE examples**

| **Example 1** | Operation | User ADM2 wants to remove RACF protection from the terminals protected by
|              |           | the generic profile $TERM$.*
|              | Known    | User ADM2 has the SPECIAL attribute. User ADM2 wants to issue the
|              |          | command as a RACF TSO command.
|              | Command  | RDELETE TERMINAL $TERM$*
|              | Defaults | None.

| **Example 2** | Operation | User JHT01 wants to remove RACF protection from the tape volume set
|              |           | VOL001.
|              | Known    | User JHT01 has the SPECIAL attribute. User JHT01 wants to issue the
|              |          | command as a RACF operator command, and the RACF subsystem prefix is @.
|              | Command  | @RDELETE TAPEVOL VOL001
|              | Defaults | None. |
## RDELETE

### Table 86. RDELETE examples (continued)

#### Example 3

**Operation** User ADM1 wants to remove the generic profile T* from the TIMS class.

**Known** User ADM1 has the SPECIAL attribute. User ADM1 wants to issue the command as a RACF TSO command.

**Command** `RDELETE TIMS T*`

**Defaults** None.

#### Example 4

**Operation** User ADM1 wants to delete the TERMINAL profiles in the RACLIST class from the RACF database and stop using RACLIST processing with the TERMINAL class. User ADM1 wants to direct the command to run at the node MVSFL under the authority of user JCARTER and prohibit the command from being automatically directed to other nodes.

**Known** Users ADM1 and JCARTER at MVSFL have the SPECIAL attribute. Users ADM1 and JCARTER at MVSFL have an already established user ID association. User ADM1 wants to issue the command as a RACF TSO command.

**Command** `RDELETE RACLIST TERMINAL ONLYAT(MVSFL.JCARTER)`

**Results** The command is only run at node MVSFL and not automatically directed to any other nodes in the RRSF configuration.
**REMOVE (Remove user from group)**

**Purpose**
You can use the REMOVE command to remove a user from a group, and to assign a new owner to any group data set profiles the user owns on behalf of that group.

**Issuing options**
The following table identifies the eligible options for issuing the REMOVE command:

<table>
<thead>
<tr>
<th>As a RACF TSO command?</th>
<th>As a RACF operator command?</th>
<th>With command direction?</th>
<th>With automatic command direction?</th>
<th>From the RACF parameter library?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

For information on issuing this command as a RACF TSO command, refer to [Chapter 3, “RACF TSO commands,” on page 15](#).

For information on issuing this command as a RACF operator command, refer to [Chapter 4, “RACF operator commands,” on page 21](#).

You must be logged on to the console to issue this command as a RACF operator command.

**Related commands**
- To add a group profile, see [ADDGROUP (Add group profile)](#).
- To change a group profile, see [ALTGROUP (Alter group profile)](#).
- To connect a user to a group, see [CONNECT (Connect user to group)](#).
- To delete a group profile, see [DELGROUP (Delete group profile)](#).
- To list a group profile, see [LISTGRP (List group profile)](#).
- To display information from a user profile, see [LISTUSER (List user profile)](#).

**Authorization required**
When issuing the REMOVE command as a RACF operator command, you might require sufficient authority to the proper resource in the OPERCMDS class. See [z/OS Security Server RACF Security Administrator’s Guide](#) for further information.

To use the REMOVE command, one of the following conditions must be true:
- You have the SPECIAL attribute.
- The group profile is within the scope of a group in which you have the group-SPECIAL attribute.
- You are the owner of the group.
- You have JOIN or CONNECT authority in the group.

To specify the AT keyword, you must have READ authority to the DIRECT.node resource in the RRSFDATA class and a user ID association must be established between the specified node.userid pair(s).
REMOVE

To specify the ONLYAT keyword you must have the SPECIAL attribute, the userid specified on the ONLYAT keyword must have the SPECIAL attribute, and a user ID association must be established between the specified node.userid pair(s) if the user IDs are not identical.

Notes:
1. If you only have ownership of the user’s profile, you do not have sufficient authority to remove the user from a group.
2. If a user is deleted from a RACF group as a result of a REMOVE command while the user is logged on, the user must logoff and logon again before that authority to access resources in classes that have been RACLISTERed is revoked. In addition, started tasks have to STOP and START to revoke the authority. This might include started tasks such as JES2 or JES3.

Syntax

For the key to the symbols used in the command syntax diagrams, see “Syntax of RACF commands and operands” on page 9. The complete syntax of the REMOVE command is:

```
[subsystem-prefix]{REMOVE | RE}
(userid ...)
[ AT([node].userid ... ) | ONLYAT([node].userid ... ) ]
[ GROUP(group-name) ]
[ OWNER(userid or group-name) ]
```

For information on issuing this command as a RACF TSO command, refer to Chapter 3, "RACF TSO commands," on page 15.

For information on issuing this command as a RACF operator command, refer to Chapter 4, "RACF operator commands," on page 21.

Parameters

**subsystem-prefix**
Specifies that the RACF subsystem is the processing environment of the command. The subsystem prefix can be either the installation-defined prefix for RACF (1–8 characters) or, if no prefix has been defined, the RACF subsystem name followed by a blank. If the command prefix was registered with CPF, you can use the MVS command D OPDATA to display it or you can contact your RACF security administrator.

Only specify the subsystem prefix when issuing this command as a RACF operator command. The subsystem prefix is required when issuing RACF operator commands.

**userid**
Specifies the user you want to remove from the group. If you are removing more than one user from the group, you must enclose the list of user IDs in parentheses.

This value is required and must be the first operand following REMOVE.

**AT | ONLYAT**
The AT and ONLYAT keywords are only valid when the command is issued as a RACF TSO command.
AT([node].userid ...)  
Specifies that the command is to be directed to the node specified by node, where it runs under the authority of the user specified by userid in the RACF subsystem address space.

If node is not specified, the command is directed to the local node.

ONLYAT([node].userid ...)  
Specifies that the command is to be directed only to the node specified by node where it runs under the authority of the user specified by userid in the RACF subsystem address space.

If node is not specified, the command is directed only to the local node.

GROUP(group-name) 
Specifies the group from which the user is to be removed. If you omit this operand, the default is your current connect group. The value specified for group-name cannot be the name of user’s default group.

OWNER(userid or group-name) 
Specifies a RACF-defined user or group that owns the group data set profiles now owned by the user to be removed.

If you omit this operand when group data set profiles exist that require a new owner, RACF does not remove the user from the group. (Group data set profiles are data set profiles whose names are qualified by the group name or begin with the value supplied by an installation exit.)

The new owner of the group data set profiles must have at least USE authority in the specified group. Do not specify a user who is being removed from the group as the new data set profile owner.

Examples

Table 87. REMOVE examples

Example 1  
Operation  User ELVIS wants to remove users KURT and JIMI from group PAYROLL.  
Known  User ELVIS has JOIN authority to group PAYROLL.

User ELVIS is currently connected to group PAYROLL.

Users KURT and JIMI are connected to group PAYROLL but do not own any group data set profiles, and group PAYROLL is not their default group.

User ELVIS wants to issue the command as a RACF TSO command.

Command  REMOVE (KURT JIMI)

Defaults  GROUP(PAYROLL)

Example 2  
Operation  User WRH0 wants to remove user PDJ6 from group RESEARCH, assigning user DAF0 as the new owner of PDJ6’s group data set profiles.  
Known  User WRH0 has CONNECT authority to group RESEARCH.

User WRH0 is not logged on to group RESEARCH.

User PDJ6 is connected to group RESEARCH and owns group data set profiles (The default connect group for user PDJ6 is not RESEARCH).

User DAF0 is connected to group RESEARCH with USE authority.

User WRH0 wants to issue the command as a RACF operator command, and the RACF subsystem prefix is @.

Command  @REMOVE PDJ6 GROUP(RESEARCH) OWNER(DAF0)

Defaults  None.
RESTART

RESTART (Restart RRSF functions)

Purpose

Use the RESTART command to restart a function in the RACF subsystem address space. The RESTART command can be used after you apply maintenance and to recover from failures.

The RESTART command ends the current subtask and starts a new one. Only one function can be restarted with a single RESTART command, but that function might involve multiple subtasks.

Note: All users or applications that update the RACF database should be completed before issuing the RESTART command.

Issuing options

The following table identifies the eligible options for issuing the RESTART command:

<table>
<thead>
<tr>
<th>As a RACF TSO command?</th>
<th>As a RACF operator command?</th>
<th>With command direction?</th>
<th>With automatic command direction?</th>
<th>From the RACF parameter library?</th>
</tr>
</thead>
<tbody>
<tr>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
</tbody>
</table>

For information on issuing this command as a RACF operator command, see "Rules for entering RACF operator commands” on page 21. You must be logged on to the console to issue this command as a RACF operator command.

Related commands

- To stop the RACF subsystem address space, see "STOP (Shutdown RRSF)” on page 631.
- To restart the RACF subsystem address space after it has been stopped, use the MVS START command.

Authorization required

When issuing the RESTART command as a RACF operator command, you might require sufficient authority to the proper resource in the OPERCMDS class. See z/OS Security Server RACF Security Administrator’s Guide for further information.

Syntax

For the key to the symbols used in the command syntax diagrams, see "Syntax of RACF commands and operands" on page 9. The complete syntax of the RESTART command is:
**Parameters**

**subsystem-prefix**
- Specifies that the RACF subsystem is the processing environment of the command. The subsystem prefix can be either the installation-defined prefix for RACF (1–8 characters) or, if no prefix has been defined, the RACF subsystem name followed by a blank. If the command prefix was registered with CPF, you can use the MVS command D OPDATA to display it or you can contact your RACF security administrator.

Subsystem prefix is a required keyword for RACF operator commands.

**COMMAND | CONNECTION | MESSAGE | RACLINK | RECEIVE | SEND**
- Specifies the restartable function.

<table>
<thead>
<tr>
<th>Function keyword</th>
<th>Module name</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>COMMAND</td>
<td>IRRSSC00</td>
<td>Command handler</td>
</tr>
<tr>
<td>CONNECTION</td>
<td>IRRDDM00</td>
<td>Device driver manager</td>
</tr>
<tr>
<td></td>
<td>IRRAPPC0</td>
<td>Device drivers</td>
</tr>
<tr>
<td></td>
<td>IRRAPPC2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>IRRSSL00</td>
<td>Local node device driver</td>
</tr>
<tr>
<td></td>
<td>IRRAPPC6</td>
<td>Handshaking</td>
</tr>
<tr>
<td>MESSAGE</td>
<td>IRRSSMG0</td>
<td>Message processor</td>
</tr>
<tr>
<td>OUTPUT</td>
<td>IRRSSOP0</td>
<td>Output handler</td>
</tr>
<tr>
<td>RACLINK</td>
<td>IRRSSK00</td>
<td>RACLINK task</td>
</tr>
<tr>
<td>RECEIVE</td>
<td>IRRSSR00</td>
<td>RRSF request receiver</td>
</tr>
<tr>
<td>SEND</td>
<td>IRRSSND0</td>
<td>RRSF request sender</td>
</tr>
</tbody>
</table>

Only one function keyword can be specified on the RESTART command at a time. If multiple function keywords are entered, the first one is processed and the rest is ignored.

The RESTART command signals the function specified by the function keyword to end and then restart.

**NODE(nodename | *)**
- The NODE operand only applies to the CONNECTION keyword. If NODE is specified for a function other than CONNECTION, it is ignored.

For information on issuing this command as a RACF operator command, see the "Rules for entering RACF operator commands" on page 21. You must be logged on to the console to issue this command as a RACF operator command.
Only a single RRSF nodename or * must be provided. Specifying * causes all device drivers to be ended and restarted. If a single nodename is provided, only the device drivers for that node are restarted.

If the NODE keyword is not specified with CONNECTION, the entire APPC server task, which includes all device drivers, is ended and restarted. As part of this processing, the device driver manager, device driver, and handshaking modules are reloaded: IRRDDM00, IRRAPPC0, IRRAPPC2, IRRAPPC6, and IRRSSL00. When the NODE keyword is specified, the device driver is ended and restarted, but no modules are reloaded.

SYSNAME(sysname | *)

restarts the connection to the specific member system sysname on the multisystem node nodename. The SYSNAME keyword is positional and must follow the NODE keyword. Specifying * causes all device drivers for each system in the particular node to end and restart.

Examples

Table 88. RESTART examples

Example 1
Operation A maintenance PTF has been applied to module IRRSSC01 and user DNP2 wants to load a new copy to put the change in effect.
Known Module IRRSSC01 resides in load module IRRSSC00.
Command @RESTART COMMAND
Defaults None.
Results The RACF subsystem prefix is @.

Example 2
Operation Restart the connection to the single-system node NODE1
Known NODE1 is a single-system node. If it were not a single-system node, RACF would issue an error message and not execute the command.
Command RESTART CONNECTION NODE (NODE1)
Defaults None.
Results The command restarts the connection to NODE1

Example 3
Operation Restart the connections to all single-system nodes, and to all member systems of multisystem nodes
Command RESTART CONNECTION NODE (*)
or RESTART CONNECTION NODE (*) SYSNAME(*)
Defaults None.
Results The command restarts the connection to all single-system nodes and to all member systems of multisystem nodes.

Example 4
Operation Restart the connections to the specific member system SYS1 on the multisystem node MULTNODE.
Command RESTART CONNECTION NODE (MULTNODE) SYSNAME(SYS1)
Defaults None.
Results The command restarts the connection to SYS1 on MULTNODE. If MULTNODE is a single-system node, RACF issues an error message.

Example 5
Operation Restart the connections to all member systems of the multisystem node MULTNODE.
Command RESTART CONNECTION NODE (MULTNODE) SYSNAME(*)
Defaults None.
Results The command restarts the connection to all member systems of MULTNODE. If MULTNODE is a single-system node, RACF issues an error message.
RLIST (List general resource profile)

Purpose

Use the RLIST command to display information on resources belonging to classes specified in the class descriptor table. Note that the DATASET, USER, and GROUP classes are not defined in the class descriptor table.

**Note:** The RLIST command might provide unpredictable results when searching on the DIGTCERT and DIGTRING classes. Due to the lowercase characters in these classes, the profile filter on the RLIST command might not function correctly.

RACF uses the class descriptor table to determine if a class is defined to RACF, the syntax of resource names within the class, and whether the class is a resource grouping class.

Profiles are listed in alphabetical order. Generic profiles are listed in the same order as they are searched for a resource match. (This also applies to the names in the global access table.)

**RACF date handling:** RACF interprets dates with 2-digit years as follows. (The yy value represents the 2-digit year.)

- If $70 < yy <= 99$, the date is interpreted as $19yy$.
- If $00 <= yy <= 70$, the date is interpreted as $20yy$.

Issuing options

The following table identifies the eligible options for issuing the RLIST command:

<table>
<thead>
<tr>
<th>As a RACF TSO command?</th>
<th>As a RACF operator command?</th>
<th>With command direction?</th>
<th>With automatic command direction?</th>
<th>From the RACF parameter library?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>

For information on issuing this command as a RACF TSO command, refer to Chapter 3, “RACF TSO commands,” on page 15.

For information on issuing this command as a RACF operator command, refer to Chapter 4, “RACF operator commands,” on page 21.

You must be logged on to the console to issue this command as a RACF operator command.

Related commands

- To list a data set profile, see LISTDSD (List data set profile).
- To list a user profile, see LISTUSER (List user profile).
- To list a group profile, see LISTGRP (List group profile).
- To obtain a list of general resource profiles, see SEARCH (Search RACF database).
Details listed

This command lists the information in an existing profile for the resource or resource group.

The details that are given for each profile are:

- The resource class.
- The name of the resource.
- The cross-reference class name (that is, the member class name for resource groups or the group name for non-group resources).
- If the resource named in the command (in the resource-name operand) is a resource group, RACF lists member resources.
- The level of the resource.
- The owner of the resource.
- The type of access attempts (as specified by the AUDIT operand on the RDEFINE or RALTER command) that are being logged on the SMF data set.
- The user, if any, to be notified when RACF uses this profile to deny access to the resource.
- The universal access authority for the resource.
- Your highest level of access authority to the resource.
- The installation-defined data (information specified in the DATA operand of the RALTER or RDEFINE commands).

If your z/OS installation is configured to be a multilevel-secure environment, this information is not listed in your output. *SUPPRESSED* appears under the installation data field. Only those with SPECIAL are allowed to list the field.

- The APPLDATA value, if any.

If your z/OS installation is configured to be a multilevel-secure environment, this information is not listed in your output. *SUPPRESSED* appears under the installation data field. Only those with SPECIAL are allowed to list the field.

- The domain distinguished name, options and local registry for the EIM segment.
- The type of access attempts (as specified by the GLOBALAUDIT operand on the RALTER command) that RACF logs.
- The status of the WARNING/NOWARNING indicator.
- For resources in the TAPEVOL class:
  - The volumes in a tape volume set,
  - Whether the TAPEVOL profile is automatic or nonautomatic,
  - Whether the volume can hold more than one data set, or
  - Whether the volume contains a TVTOC.

Additional details:

You can request the following details by using the appropriate RLIST operands:

- The security label, the security level and categories.
  For additional information, see the AUTHUSER operand.
- For member resources, RACF lists the names of all resource group members in which the entity is a member.
  For additional information, see the RESGROUP operand.
- The number of times the resource was accessed by all users for each of the following access authorities.
– ALTER, CONTROL, UPDATE, READ

For additional information, see the STATISTICS operand. This detail is only meaningful when your installation is gathering resource statistics and the class is not RACTIONed. For a generic profile, RACF replaces any statistics line with NOT APPLICABLE FOR GENERIC PROFILE.

• Historical data, such as:
  – Date the resource was defined to RACF,
  – Date the resource was last referenced (this detail is only meaningful when your installation is gathering resource statistics and the class is not RACTIONed; for a generic profile, RACF replaces any statistics line with NOT APPLICABLE FOR GENERIC PROFILE), or
  – Date the resource was last accessed at the update level.
For additional information, see the HISTORY operand.

• The standard access list which displays:
  – All users and groups authorized to access the resource,
  – The level of authority for each user and group, or
  – The number of times each user has accessed the resource. (This detail is only meaningful when your installation is gathering resource statistics. This detail is not included in the output for generic profiles.)
For additional information, see the AUTHUSER operand.

• The conditional access list which displays the same fields as the standard access list, as well as the following additional fields:
  – The class of the resource, or
  – The entity name of the resource.
For additional information, see the AUTHUSER operand.

• For a tape volume that contains RACF-protected data sets, the following information about each RACF-protected data set on the volume:
  – The name used to create the data set,
  – The internal RACF name for the data set,
  – The volumes on which the data set resides,
  – The file sequence number for the data set,
  – The date when the data set was created, or
  – Whether the data set profile is discrete or generic.
For additional information, see the TVTOC operand.

• The contents of segments other than the base segment...
(See the segment operands for details about the listed information.)

**Authorization required**

When issuing the RLIST command as a RACF operator command, you might require sufficient authority to the proper resource in the OPERCMDS class. See [z/OS Security Server RACF Security Administrator's Guide](https://www.ibm.com/support/knowledgecenter/SSHKQY_7.3.0/com.ibm.zosКАС/gls/ksclm00.htm#7.9.4) for further information.

You must have a sufficient level of authority for each resource or resource group listed as the result of your request so that one of the following conditions is met:

• You have the SPECIAL attribute.
• The resource profile is within the scope of a group in which you have the group-SPECIAL attribute.
• You have the OPERATIONS attribute.
The resource profile is within the scope of a group in which you have the group-OPERATIONS attribute.

You have the AUDITOR attribute.

You are the owner of the resource.

If the profile is in the FILE or DIRECTRY class, the second qualifier of the profile name is your user ID.

To list the contents of segments other than the base segment, such as DLFDATA, you must have the SPECIAL attribute or your installation must permit you to do so through field-level access checking.

You are on the access list for the resource and you have at least READ authority. (If your level of authority is NONE, the resource is not listed.) If you specify ALL, RACF lists only information pertinent to your user ID.

Your current connect group (or, if list-of-groups checking is active, any group to which you are connected) is in the access list and has at least READ authority.

The universal access authority of the resource is at least READ.

You have at least read access for the profile name from the GLOBAL ENTRY TABLE (if this table contains an entry for the profile).

You see the type of access attempts, as specified by the GLOBALAUDIT operand, only if you have the AUDITOR attribute or if the resource profile is within the scope of a group in which you have the group-AUDITOR attribute.

To specify the AT keyword, you must have READ authority to the DIRECT.node resource in the RRSFDATA class and a user ID association must be established between the specified node.userid pair(s).

To specify the ONLYAT keyword you must have the SPECIAL attribute, the userid specified on the ONLYAT keyword must have the SPECIAL attribute, and a user ID association must be established between the specified node.userid pair(s) if the user IDs are not identical.

**Listing resource access lists:** When you are requesting to see the access list for a resource with the AUTHUSER operand, your level of authority is checked for each resource. Your level of authority must be such that one of the following conditions is met:

- You have the SPECIAL attribute.
- The resource profile is within the scope of a group in which you have the group-SPECIAL attribute.
- You have the OPERATIONS attribute.
- The resource profile is within the scope of a group in which you have the group-OPERATIONS attribute.
- You are the owner of the resource.
- You have the AUDITOR attribute.
- The resource profile is within the scope of a group in which you have the group-AUDITOR attribute.
- You have alter access for the profile name from the GLOBAL ENTRY TABLE (if this table contains an entry for the profile).
- If the profile is in the FILE or DIRECTRY class, the second qualifier of the profile name is your user ID.
For a discrete profile, you are on the access list for the resource and you have ALTER authority. (If you have any other level of authority, you cannot use the operand.)

For a discrete profile, your current connect group (or, if list-of-groups checking is active, any group to which you are connected) is in the access list and has ALTER authority.

For a discrete profile, the universal access authority of the resource is ALTER.

Syntax

For the key to the symbols used in the command syntax diagrams, see “Syntax of RACF commands and operands” on page 9. The complete syntax of the RLIST command is:

```
[subsystem-prefix][RLIST | RL]
  class-name
  (profile-name ...) | *
  [ ALL ]
  [ AT([node].userid ...) | ONLYAT([node].userid ...) ]
  [ AUTHUSER ]
  [ CDTINFO ]
  [ CFDEF ]
  [ DLFDATA ]
  [ EIM ]
  [ [GENERIC | NOGENERIC] ]
  [ HISTORY ]
  [ ICSF ]
  [ ICTX ]
  [ KERB ]
  [ NORACF ]
  [ NOYOURACC ]
  [ PROXY]
  [ RESGROUP ]
  [ SESSION ]
  [ SIGVER ]
  [ SSIGNON ]
  [ STATISTICS ]
  [ STDATA ]
  [ SVFMR ]
  [ TME ]
  [ TVTOC ]
```

For information on issuing this command as a RACF TSO command, refer to Chapter 3, “RACF TSO commands,” on page 15.

For information on issuing this command as a RACF operator command, refer to Chapter 4, “RACF operator commands,” on page 21.

Parameters

`subsystem-prefix`

Specifies that the RACF subsystem is the processing environment of the command. The subsystem prefix can be either the installation-defined prefix for RACF (1–8 characters) or, if no prefix has been defined, the RACF subsystem.
name followed by a blank. If the command prefix was registered with CPF, you can use the MVS command D OPDATA to display it or you can contact your RACF security administrator.

Only specify the subsystem prefix when issuing this command as a RACF operator command. The subsystem prefix is required when issuing RACF operator commands.

class-name
Specifies the name of the class to which the resource belongs. Valid class names are those specified in the class descriptor table. For a list of general resource classes defined in the class descriptor table supplied by IBM, see Appendix B, “Supplied RACF resource classes,” on page 659.

This operand is required and must be the first operand following RLIST.

This command is not intended to be used for profiles in the following classes:
- DCEUUIDS
- DIGTCERT
- DIGTNMAP
- DIGTRING
- IDIDMAP
- NDSLINK
- NOTELINK
- ROLE
- UNIXMAP.

(profile-name ...) | *

(profile-name ...)
Specifies the name of an existing discrete or generic profile about which information is to be displayed.

The variable profile-name or an asterisk (*) is required and must be the second operand following RLIST.

If you specify more than one value for profile-name, the list of names must be enclosed in parentheses.

Mixed-case profile names are accepted and preserved when class-name refers to a class defined in the static class descriptor table with CASE=ASIS or in the dynamic class descriptor table with CASE(ASIS).

If the resource specified is a tape volume serial number that is a member of a tape volume set, information on all the volumes in the set are displayed.

RACF processes each resource you specify independently. If an error occurs while processing a resource, RACF issues a message and continues processing with the next resource.

Note: Inactive SECLABEL profiles and profiles that contain inactive security labels may not be listed if SETROPTS SECLBYSYSTEM is active because only users with SPECIAL or AUDITOR authority are allowed to view inactive security labels.

* Specifies that you want to display information for all resources defined to the specified class for which you have the proper authority.

On a system with many profiles defined, the use of * may result in a large amount of output that may not be useful to a user issuing the command. It
may be more appropriate for the user to browse the output of IRRDBU00 (database unload) or to write a program to process the IRRDBU00 output and produce a report showing only the subset of information that is of interest to the user. The processing of output of RLIST by programs is not supported nor recommended by IBM. If you want a listing of all the profiles for use by a program you should instead have the program process the output from IRRDBU00, RACROUTE REQUEST=EXTRACT, or ICHEINTY.

An asterisk (*) or profile-name is required and must be the second operand following RLIST.

RACF processes each resource independently and displays information only for those resources for which you have sufficient authority.

If you have the AUDITOR attribute, or if the resource profile is within the scope of a group in which you have the group-AUDITOR attribute, RACF displays GLOBALAUDIT information for all resources in the class.

ALL

Specifies that you want all information for the BASE segment of each resource displayed.

The access list is included only if you have sufficient authority to use the AUTHUSER operand. (See Authorization required.) The type of access attempts (as specified by the GLOBALAUDIT operand) that are being logged on the SMF data set is included only if you have the AUDITOR attribute, or the resource profile is within the scope of a group in which you have the group-AUDITOR attribute.

AT | ONLYAT

The AT and ONLYAT keywords are only valid when the command is issued as a RACF TSO command.

AT([node].userid ...)

Specifies that the command is to be directed to the node specified by node, where it runs under the authority of the user specified by userid in the RACF subsystem address space.

If node is not specified, the command is directed to the local node.

ONLYAT([node].userid ...)

RLIST is not eligible for automatic command direction. If you specify the ONLYAT keyword, the effect is the same as if you specified the AT keyword.

AUTHUSER

Specifies that you want the following information included in the output:

- The user categories authorized to access the resource
- The security level required to access the resource
- The security label required to access the resource
- The standard access list. This includes the following:
  - All users and groups authorized to access the resource
  - The level of authority for each user and group
  - The number of times the user has accessed the resource (This detail is only meaningful when your installation is gathering resource statistics and is not included in the output for generic profiles.)
- The conditional access list. This list consists of the same fields as in the standard access list, as well as the following fields:
RLIST

- The class of the resource through which each user and group in the list can access the target resource of the command. For example, if a user can access the target resource through terminal TERM01, then TERMINAL would be the class listed.

- The entity name of the resource through which each user and group in the list can access the target resource of the command. In the example above, TERM01 would be listed.

You must have sufficient authorization to use the AUTHUSER operand. (See Authorization required.)

CDTINFO
Specifies that CDTINFO segment information should be listed for profiles in the CDT class.

CFDEF
Specifies that CFDEF segment information should be listed for profiles in the CFIELD class. Use this operand to display the custom field names and attributes, such as data type, that your installation has defined.

Contact your security administrator to see how custom fields are used at your installation. For more information about custom fields, see z/OS Security Server RACF Security Administrator's Guide.

DLFDATA
Lists the contents of the DLFDATA segment for profiles in the DLFCLASS class.

EIM
Specifies that EIM segment information should be listed.

GENERIC | NOGENERIC

GENERIC
Specifies that you want RACF to display information for the generic profile that most closely matches a resource name. If you specify GENERIC, RACF ignores a discrete profile that protects the resource. If asterisk (*) is specified instead of the profile name, all generic profiles are listed.

NOGENERIC
Specifies that you want RACF to display information for the discrete profile that protects a resource. If asterisk (*) is specified instead of the profile name, all discrete profiles are listed.

If neither GENERIC nor NOGENERIC is specified, RACF lists information for the discrete resource name that matches the resource name you specify. If there is no matching discrete profile, RACF lists the generic profile that most closely matches the resource name. If asterisk (*) is specified instead of the profile name, all discrete and generic profiles are listed.

The following list shows examples of using the GENERIC and NOGENERIC operands:

- If you enter the following command, RACF lists all discrete and generic profiles in the DASDVOL class.
  RLIST DASDVOL *

- If you enter the following command, RACF lists information for all the generic profiles in the DASDVOL class.
  RLIST DASDVOL * GENERIC

- If you enter the following command, RACF lists all discrete profiles in the JESSPOOL class.
If you enter the following command, RACF displays the best-fit generic profile that protects the resource ABC.DEF. RACF ignores discrete profile ABC.DEF if it exists.

RLIST APPCLU ABC.DEF GENERIC

**Note:** When searching for a generic profile that matches the specified resource, RACF does not examine members that are defined in a grouping class (through the ADDMEM operand of the RDEFINE command). For example, suppose two profiles had been defined by the following RDEFINE commands:

```
RDEFINE TCICSTRN A *
RDEFINE GCICSTRN xxx ADDMEM(AB *)
```

The command:

```
RLIST TCICSTRN ABC
```

displays profile A * in the TCICSTRN class, but it does not search the GCICSTRN class and therefore does not display any AB* profile of the GCICSTRN class. In addition, the command:

```
RLIST GCICSTRN ABC
```

does not find member AB* in the GCICSTRN class because it does not look at the members in a grouping class.

If you wish to make use of RLIST to find the generic profile that protects a specific resource, and the resource is in a class that has both a grouping class and a member class, you should define the generic profile as a profile in the member class.

To illustrate the above RDEFINE example where ADDMEM(AB*) had been specified for a grouping class, the following command:

```
RDEFINE TCICSTRN AB *
```

allows the RLIST command to display AB* as the generic member in the TCICSTRN class.

**HISTORY**

Specifies that you want to list the following data:

- The date each profile was defined to RACF
- The date each profile was last referenced (this detail is only meaningful when your installation is gathering resource statistics; for a generic profile and profiles that are RAONLYed, RACF replaces any statistics line with NOT APPLICABLE FOR GENERIC PROFILE)
- The date of last RACROUTE REQUEST=AUTH for UPDATE authority (this detail is only meaningful when your installation is gathering resource statistics; for a generic profile and profiles that are RAONLYed, RACF replaces any statistics line with NOT APPLICABLE FOR GENERIC PROFILE)

**ICSF**

Specifies that ICSF segment information should be listed for profiles in the CSFKEYS, GCSFKEYS, XCSFKEY, or GXCSFKEY class.

**ICTX**

Specifies that ICTX segment information should be listed.
KERB
Specifies that you want to list the following z/OS Integrated Security Services
Network Authentication Service information:
• The local kerberos-realm-name (KERBNAME)
• The encryption value settings (ENCRYPT values or NOENCRYPT)
• The min-ticket-life value for the local realm (MINTKTLFE)
• The def-ticket-life value for the local realm (DEFTKTLFE)
• The max-ticket-life value for the local realm (MAXTKTLFE)
• The current key version (KEY VERSION)

Note: If KEY VERSION is not displayed, there is no z/OS Network
Authentication Service key associated with this realm definition.

NORACF
Specifies that you want to suppress the listing of RACF segment information. If
you specify NORACF, you must include either CDTINFO, DLFDATA, EIM,
KERB, PROXY, SESSION, SSIGNON, STDATA, SVFMR, TME, or a
combination of operands.
If you do not specify NORACF, RACF displays the information in the base
segment of a general resource profile.
The information displayed as a result of using the NORACF operand is
dependent on other operands used in the command. For example, if you use
NORACF with SESSION also specified, only the SESSION information is
displayed.

NOYOURACC
For grouping and member classes, RLIST must do additional processing to
assure that the your access information field is accurate. A SPECIAL user can
use the NOYOURACC operand to bypass this processing, for performance
reasons. The your access field contains n/a in this circumstance.

Note: This operand applies to SPECIAL users only. It has no effect for other
users.

PROXY
Specifies that PROXY segment information should be listed. The following
information will be provided:
• the URL of the LDAP server to be contacted
• the BIND distinguished name
• information regarding the BIND password
  The BINDPW password values will not be listed. If a BINDPW password
value is defined for a general resource profile, RLIST will display YES for the
PROXY segment BINDPW attribute. If no BINDPW password value has been
defined, RLIST will display NO for the PROXY segment BINDPW attribute.

RESGROUP
Requests a list of all resource groups of which the resource specified by the
profile-name operand is a member.
If a profile does not exist for the specified resource, RACF lists the names of all
resource groups of which the resource is a member and to which the command
user is authorized. To be authorized, the command user must meet one of the
authorization requirements listed in Authorization required.
If a profile *does* exist for the specified resource and the command user has ALTER authority to the resource, RACF lists the names of all groups of which the resource is a member.

If a profile *does* exist for the specified resource but the command user has less than ALTER authority to the resource, RACF lists the names of all groups of which the resource is a member and to which the command user is authorized. To be authorized to the resource group, the command user must meet one of the authorization requirements listed in [Authorization required]. However, the command issuer must have the authority to list the resource specified on the command in order to list the member groups. If this requirement is met, then the user must be also authorized to the resource group. Otherwise, an error message is issued.

When `profile-name` is the name of a protected resource (such as a terminal or DASD volume) and `class-name` is a member class (such as TERMINAL or DASDVOL), the RESGROUP operand lists the profiles that protect the resource (for example, profiles in the GTERMINL or GDASDVOL class).

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

```
RDEF GIMS GIMSGRP ADDMEM(ABC*)
```

and you are looking for a specific member, so you enter:

```
RLIST GIMS ABCD RESGROUP
```

`ABC*` will not appear in the output.

**Note:** When considering this example, if you are unable to define the profile ABCD, it might be due to a generic definition somewhere in GIMS.

This operand applies only to member classes for which resource group profiles exist.

**SESSION**

Specifies that the contents of the SESSION segment are to be listed for profiles in the APPCLU class.

**SIGVER**

Specifies that the contents of the SIGVER segment are to be listed for profiles in the PROGRAM class.

**SSIGNON**

Specifies that you want to display the secured signon information.

**Note:** The secured signon application key value cannot be displayed. However, information is displayed that describes whether the key value is masked or encrypted.

**STATISTICS**

Specifies that you want to list the statistics for each resource. The list contains the number of times the resource was accessed by users with READ, UPDATE, CONTROL, and ALTER authorities. A separate total is given for each authority level.

**Note:** This detail is only meaningful when your installation is gathering resource statistics. For a generic profile, RACF replaces any statistics line with NOT APPLICABLE FOR GENERIC PROFILE.
### RLIST

**STDATA**
- Specifies that you want to list the contents of the STDATA segment for profiles in the STARTED class.

**SVFMR**
- Lists the contents of the SVFMR segment for profiles in the SYSMVIEW class.

**TME**
- Specifies that information in the Tivoli Security Management Application is to be listed.

**TVTOC**
- Specifies that you want to see information about the data sets defined in the TVTOC of a TAPEVOL profile. The output displays:
  - The name used to create the data set
  - The internal RACF name for the data set
  - The volumes on which the data set resides
  - The file sequence number for the data set
  - The date when the data set was created
  - Whether the data set profile is discrete or generic.

### Examples

**Table 89. RLIST examples**

#### Example 1
**Operation**  
User RV2 wants to list all information about the tape volume VOL001.

**Known**  
User RV2 is the owner of tape volume VOL001.

  - User RV2 has the AUDITOR attribute.
  - User RV2 wants to issue the command as a RACF TSO command.

**Command**  
```plaintext
RLIST TAPEVOL VOL001 ALL
```

**Defaults**  
None.

**Output**  
See Figure 52.

#### Example 2
**Operation**  
User ADM1 wants to list information about the generic profile T* in the TIMS class.

**Known**  
User ADM1 has the SPECIAL and AUDITOR attributes.

  - User ADM1 wants to issue the command as a RACF TSO command.

**Command**  
```plaintext
RLIST TIMS T*
```

**Defaults**  
None.

**Output**  
See Figure 53.

#### Example 3
**Operation**  
User IBMUSER wants to list information about the profile TERM1 in the TERMINAL class. TERM1 is a member of four GTERMINL class profiles: GTERM1, GTERM2, GTERM3, and GTERM4. TERM1 has a UACC of NONE.

**Known**  
User IBMUSER has the SPECIAL and AUDITOR attributes. User IBMUSER wants to issue the command as a RACF TSO command.

**Command**  
```plaintext
RLIST TERMINAL TERM1 RESGROUP
```

**Defaults**  
None.

**Output**  
See Figure 54.
Table 89. RLIST examples (continued)

Example 4

**Operation**
The security administrator wants to display secured signon key information for profile name TSOR001 in the PTKTDATA class to be certain that the application key is masked instead of encrypted.

**Known**
ELVIS1 is the user ID of the security administrator and has the SPECIAL attribute. The security administrator wants to issue the command as a RACF TSO command.

**Command**
RLIST PTKTDATA TSOR001 SSIGNON

**Defaults**
None.

**Output**
See Figure 55.

Example 5

**Operation**
The security administrator wants to display secured signon key information for profile name TSOR004 in the PTKTDATA class and to be certain that the application key is encrypted instead of masked.

**Known**
NONNEL is the user ID of the security administrator and has the SPECIAL attribute. The security administrator wants to issue the command as a RACF operator command, and the RACF subsystem prefix is @.

**Command**
@RLIST PTKTDATA TSOR004 SSIGNON

**Defaults**
None.

**Output**
See Figure 56.

Example 6

**Operation**
The security administrator wants to display the contents of the STDATA segments for profiles in the STARTED class with the generic profile name (VTAML.*).

**Known**
SYSUSER is the user ID of the security administrator and has the SPECIAL attribute. The security administrator wants to issue the command as a RACF TSO command.

**Command**
RLIST STARTED VTAM.* STDATA NORACF

**Defaults**
None.

**Output**
See Figure 57.

Example 7

**Operation**
The security administrator wants to list the contents of the KERBDFLT profile in the REALM class.

**Known**
The administrator has access to the KERBDFLT profile in the REALM class.

**Command**
RLIST REALM KERBDFLT KERB NORACF

**Defaults**
None.

**Output**
See Figure 58.

Example 8

**Operation**
The administrator wants to list the contents of a profile (TSOIM13) in the PTKTDATA class. This particular PassTicket profile indicates that replay protection is to be bypassed.

**Known**
The administrator has access to the PTKTDATA class.

**Command**
RLIST PTKTDATA TSOIM13

**Defaults**
None.

**Output**
See Figure 59.

Example 9

**Operation**
The administrator wants to list the contents of a profile (IRR.PROXY.DEFAULTS) in the FACILITY class and the contents of the EIM segment. This particular PROXY profile indicates that a BINDPW has been defined.

**Known**
The administrator has access to the FACILITY class.

**Command**
RLIST FACILITY IRR.PROXY.DEFAULTS EIM PROXY NORACF

**Defaults**
None.

**Output**
See Figure 60.

Example 10

**Operation**
The security administrator wants to list class descriptor table (CDT) information of the TSTCLAS8 in the CDT class.

**Known**
The administrator has the SPECIAL attribute.

**Command**
RLIST CDT TSTCLAS8 NORACF CDTINFO

**Defaults**
None.

**Output**
See Figure 61.
**Table 89. RLIST examples (continued)**

**Example 11**
**Operation** The security administrator Rui wants to list the contents of the IRR.ICTX.DEFAULTS profile in the LDAPBIND class and the contents of the ICTX segment.

**Known** Rui has READ access to the LDAPBIND class.

**Command** RLIST LDAPBIND IRR.ICTX.DEFAULTS ICTX NORACF

**Defaults** None.

**Output** See Figure 62.

**Example 12**
**Operation** Rui wants to list the access list for the DSN.ZHAOHUI.TABLE.ALTER resource in the MDSNTB class.

**Known** Rui has the SPECIAL attribute.

**Command** RLIST MDSNTB DSN.ZHAOHUI.TABLE.ALTER AUTHUSER

**Defaults** None.

**Output** See Figure 63.

**Example 13**
**Operation** The security administrator uses the custom field named EMPSER for employee serial numbers. She wants to list the attributes of this custom field for user profiles.

**Known** The security administrator has the SPECIAL attribute.

**Command** RLIST CFIELD USER.CSDATA.EMPSER CFDEF NORACF

**Defaults** None.

**Output** See Figure 64.

**Example 14**
**Operation** The security administrator wants to list the settings related to digital signature verification for the program called XYZLIB64.

**Known** The security administrator has the SPECIAL attribute.

**Command** RLIST PROGRAM XYZLIB64 SIGVER NORACF

**Defaults** None.

**Output** See Figure 65.

**Example 15**
**Operation** The security administrator wants to list ICSF segment information for all profiles in the XCSFKEY class.

**Known** The security administrator has the SPECIAL attribute.

**Command** RLIST XCSFKEY * ICSF NORACF

**Defaults** None.

**Output** See Figure 66.
Figure 52. Example 1: Output for the RLIST command
Figure 53. Example 2: Output for the RLIST command

Figure 54. Example 3: Output for the RLIST command with RESGROUP option
SSIGNON INFORMATION  
---------------------  
KEYMASKED DATA NOT DISPLAYABLE  

Figure 55. Example 4: Output for RLIST command with masked application key

SSIGNON INFORMATION  
---------------------  
KEYENCRIPTED DATA NOT DISPLAYABLE

Figure 56. Example 5: Output for RLIST command with encrypted application key

STDATA INFORMATION  
------ -----------  
USER= SYSUSER  
GROUP= SYSGROUP  
TRUSTED= YES  
PRIVILEGED= NO  
TRACE= NO  

Figure 57. Example 6: Output for RLIST command for the STDATA segment

CLASS NAME  
----- ----  
REALM KERBDFLT

KERB INFORMATION  
----------------  
KERBNAME= KRB2000.IBM.COM  
MINTKTLFE= 0000000300  
MAXTKTLFE= 000000086400  
DEFTKTLFE= 0000036000  
KEY VERSION= 001  
KEY ENCRYPTION TYPE= DES DES3 DESD AES128 AES256

------------------  
CLASS NAME  
----- ----  
REALM /.../KERB390.ENDICOTT.IBM.COM/KRBGTG/KER2000.ENDICOTT.IBM.COM

Figure 58. Example 7: Output for RLIST command for the KERB segment
<table>
<thead>
<tr>
<th>CLASS NAME</th>
<th>LEVEL</th>
<th>OWNER</th>
<th>UNIVERSAL ACCESS</th>
<th>YOUR ACCESS</th>
<th>WARNING</th>
</tr>
</thead>
<tbody>
<tr>
<td>PTKTDATA</td>
<td>00</td>
<td>IBMUSER</td>
<td>NONE</td>
<td>NONE</td>
<td>NO</td>
</tr>
</tbody>
</table>

Installation Data
-------------------
NONE

Application Data
-------------------
NO REPLAY PROTECTION

AUDITING
---------
FAILURES(READ)

NOTIFY
---------
NO USER TO BE NOTIFIED

Figure 59. Example 8: Output for RLIST command in the PTKTDATA class

```
RLIST FACILITY IRR.PROXY.DEFAULTS EIM NORACF
CLASS NAME
------- ----
FACILITY IRR.PROXY.DEFAULTS

EIM INFORMATION
--------------
EIM OPTIONS= ENABLE
LOCALREGISTRY= SYS1SAF
KERBREGISTRY= MYCOMPANYREALM
X509REGISTRY= MYCOMPANYCERTS
```

Figure 60. Example 9: Output for RLIST command for the EIM segment
Figure 61. Example 10: Output for RLIST command for the CDTINFO segment

Figure 62. Example 11: Output for RLIST of the ICTX segment
<table>
<thead>
<tr>
<th>Class</th>
<th>Name</th>
<th>Level</th>
<th>Owner</th>
<th>Universal Access</th>
<th>Your Access</th>
<th>Warning</th>
</tr>
</thead>
<tbody>
<tr>
<td>MDSNTB</td>
<td>DSN.ZHAOHUI.TABLE.ALTER</td>
<td>00</td>
<td>ADMRUI</td>
<td>None</td>
<td>ALTER</td>
<td>NO</td>
</tr>
</tbody>
</table>

**Installation Data**

- None

**Application Data**

- None

**Security Level**

- No Security Level

**Categories**

- No Categories

**Security Label**

- No Security Label

**Auditing**

- Failures (Read)

**Notify**

- No User to Be Notified

<table>
<thead>
<tr>
<th>User</th>
<th>Access</th>
<th>Access Count</th>
<th>Class</th>
<th>Entity Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>JEAN</td>
<td>READ</td>
<td>000000</td>
<td>CRITERIA</td>
<td>SQLROLE=TELLER</td>
</tr>
</tbody>
</table>

*Figure 63. Example 12: Output for RLIST of the AUTHUSER segment*
**Figure 64. Example 13: Output for RLIST of the CFDEF segment**

```
RLIST CFIELD USER.CSDATA.EMPSER CFDEF NORACF
CLASS     NAME
-----     ----
CFIELD    USER.CSDATA.EMPSER

CFDEF INFORMATION
-----------------
TYPE = NUM
MAXLENGTH = 00000008
MAXVALUE = 0099999999
MINVALUE = 0000100000
FIRST = NUMERIC
OTHER = NUMERIC
MIXED = NO
HELP = EMPLOYEE SERIAL NUMBER, 6-8 DIGITS
LISTHEAD = EMPLOYEE SERIAL =
```

**Figure 65. Example 14: Output for RLIST of the SIGVER segment**

```
RLIST PROGRAM XYZLIB64 SIGVER NORACF
CLASS     NAME
-----     ----
PROGRAM   XYZLIB64

SIGVER INFORMATION
------------------
SIGREQUIRED = YES
FAILLOAD = ANYBAD
SIGAUDIT = ANYBAD
```

**Figure 66. Example 15: Output for RLIST of the ICSF segment**

```
RLIST XCSFKEY * ICSF NORACF
CLASS     NAME
-----     ----
XCSFKEY   ATEST

ICSF INFORMATION
----------------
SYMEXPORTABLE = BYLIST
SYMEXPORTCERTS = DENICE/CertForDenice KEN/Cert for Ken
ASYMUSAGE = HANDSHAKE SECUREEXPORT

CLASS     NAME
-----     ----
XCSFKEY   BTEST

ICSF INFORMATION
----------------
SYMEXPORTABLE = BYLIST
SYMEXPORTCERTS = *
SYMEXPORTKEYS = PKDS.LABEL1 PKDS.LABEL2
ASYMUSAGE = HANDSHAKE SECUREEXPORT
```
RVARY (Change status of RACF database)

Purpose

Use the RVARY command to:

- Deactivate and reactivate the RACF function.
- Switch from using a specific primary data set to using its corresponding backup data set, perhaps because of a failure related to the primary data set.
- Deactivate or reactivate primary or backup RACF data sets. (Deactivating a specific primary data set causes all RACF requests for access to that data set to fail. Deactivating a specific backup data set causes RACF to stop duplicating information on that data set.)
- Deactivate protection for any resources belonging to classes defined in the class descriptor table while RACF is inactive.
- Select the mode of operation when RACF is enabled for sysplex communication.

While RACF is deactivated, utilities can be run to diagnose and repair logical errors in the RACF database. RACF installation exits can provide special handling for requests to access RACF-protected resources (for example, by prompting the operator to allow or deny access). If the RACF data set is itself RACF-protected, RACF failsoft processing, which can include installation exit routine processing, controls access to the RACF database. When you deactivate RACF using the RVARY command, only users defined in TSO SYS1.UADS can still log on to TSO, and RACF does not validate those user IDs. When RACF is inactive, failsoft processing takes effect.

**Note:** Failsoft processing occurs only when all primary RACF data sets are inactive. If you have multiple RACF data sets and only one is inactive, you are likely to experience ABENDs. See [z/OS Security Server RACF System Programmer’s Guide](https://www.ibm.com/support/docview.ws/docuid/36a15553-61e1-11e6-82a8-0010a1e8855c) for more information on failsoft processing and using RVARY.

RACF logs each use of the RVARY command provided that the system has been IPLed with RACF active and the use of RVARY changes the status of RACF. For example, if you issue RVARY to deactivate a RACF database that is already inactive, you do not change the status of RACF. Therefore, RACF does not log this particular use of RVARY. When RACF is enabled for sysplex communication, logging of the RVARY commands occurs only on the system from which the command originated.

When you deactivate a RACF data set (using RVARY INACTIVE) or switch to a backup RACF data set (using RVARY SWITCH), RACF automatically deallocates that data set. To reactivate a data set, use the RVARY ACTIVE command. The RVARY SWITCH does not activate an inactive data set. RACF automatically reallocates that data set. This feature allows you to restore the data set from a copy on tape or recatalog the data set on another volume without having to re-IPL your system.

If you deactivate the primary RACF data set, and uncatalog it, and replace it with an alternate data set, the alternate data set must be cataloged and have the same name as the original data set before you can activate it. When you deactivate (and deallocate) a RACF data set, you can move the data set from one direct access storage device to another.
Before recataloging a data set, you must first deactivate the data set by issuing either the RVARY INACTIVE or the RVARY SWITCH command.

**Using RVARY when RACF is enabled for sysplex communication:** In addition to the RVARY DATAShare and RVARY NODATAShare commands, which are valid only when RACF is enabled for sysplex communication, the following RVARY commands are propagated when RACF is enabled for sysplex communication:

- RVARY ACTIVE
- RVARY INACTIVE
- RVARY SWITCH

When issued from any member of the RACF data sharing group, these commands are propagated in a controlled, synchronized manner to each of the other members in the group.

**Notes:**

1. For RVARY INACTIVE(NOCLASSACT) and RVARY INACTIVE(NOTAPE) commands, only the RVARY INACTIVE portion of the command is propagated.
2. The MVS operator commands ROUTE *ALL and ROUTE system-group-name are allowed only with RVARY LIST.
3. RACF does not propagate commands if the system is operating in failsoft mode unless failsoft mode was entered because an RVARY INACTIVE command was issued.
4. RVARY INACTIVE DATASET, SWITCH, DATAShare, and NODATAShare require that RVARY quiesce RACF database I/O activity before proceeding. There can be no database I/O activity in progress while the status of the database is changed or the database could get corrupted. Consequently, RVARY must wait for previously scheduled database I/O to complete before proceeding. If there are problems with the DASD device the data set is on and the I/O is hung, those problems have to be cleared up before the command can complete. See the RVARY command documentation in the “Recovery Procedures” chapter of [z/OS Security Server RACF System Programmer’s Guide](https://www.ibm.com) for more information.

**Issuing options**

The following table identifies the eligible options for issuing the RVARY command:

<table>
<thead>
<tr>
<th>As a RACF TSO command?</th>
<th>As a RACF operator command?</th>
<th>With command direction?</th>
<th>With automatic command direction?</th>
<th>From the RACF parameter library?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>

For information on issuing this command as a RACF TSO command, refer to Chapter 3, “RACF TSO commands,” on page 15.

For information on issuing this command as a RACF operator command, refer to Chapter 4, “RACF operator commands,” on page 21.

You must be logged on to the console to issue this command as a RACF operator command.
RVARY

Authorization required

No special authority is needed to issue the RVARY command. However, the operator (at the operator console or security console) must approve a change in RACF status or the RACF data sets—or a change in the operational mode if RACF is enabled for sysplex communication—before RACF allows the command to complete.

If the RVARY command changes RACF or database status (ACTIVE/INACTIVE), RACF issues an informational message and the operator is required to enter the password defined by RVARYPW STATUS(status-pw) to authorize the change. If the RVARY command switches the RACF data sets (SWITCH) or changes the RACF operating mode (DATASHARE/NODATASHARE), RACF issues an informational message and the operator is required to enter the password defined by RVARYPW SWITCH(switch-pw). When RVARY is issued as a RACF operator command from a console with master authority, the default password YES is also accepted for RVARY ACTIVE, RVARY NODATASHARE or RVARY SWITCH commands.

Syntax

For the key to the symbols used in the command syntax diagrams, see “Syntax of RACF commands and operands” on page 9. The complete syntax of the RVARY command is:

```
[subsystem-prefix]RVARY
   [ ACTIVE
      | INACTIVE
      | INACTIVE NOCLASSACT(class-namelist | *)
      | INACTIVE(NOTAPE)
      | DATASHARE
      | NODATASHARE
      | SWITCH ]
   [ DATASET(data-set-name... | *) ]
   [ LIST | NOLIST ]
```

For information on issuing this command as a RACF TSO command, refer to Chapter 3, “RACF TSO commands,” on page 15.

For information on issuing this command as a RACF operator command, refer to Chapter 4, “RACF operator commands,” on page 21.

Parameters

```
[subsystem-prefix] RVARY
   [ ACTIVE
      | INACTIVE
      | INACTIVE NOCLASSACT(class-namelist | *)
      | INACTIVE(NOTAPE)
      | DATASHARE
      | NODATASHARE
      | SWITCH ]
   [ DATASET(data-set-name... | *) ]
   [ LIST | NOLIST ]
```

```
For information on issuing this command as a RACF TSO command, refer to Chapter 3, “RACF TSO commands,” on page 15.
```

```
For information on issuing this command as a RACF operator command, refer to Chapter 4, “RACF operator commands,” on page 21.
```

```
Parameters

```
subsystem-prefix
```

Specifies that the RACF subsystem is the processing environment of the command. The subsystem prefix can be either the installation-defined prefix for RACF (1–8 characters) or, if no prefix has been defined, the RACF subsystem name followed by a blank. If the command prefix was registered with CPF, you can use the MVS command D OPDATA to display it or you can contact your RACF security administrator.

Only specify the subsystem prefix when issuing this command as a RACF operator command. The subsystem prefix is required when issuing RACF operator commands.

```
ACTIVE
```

Specifies that the RACF function for, and access to, the primary RACF database is to be reactivated.
If you wish to reactivate a particular primary data set or if you wish to activate or reactivate a backup data set, then you must specify the DATASET operand with the appropriate data set name.

When you reactivate any RACF data set it is automatically reallocated.

When RACF is enabled for sysplex communication and RVARY ACTIVE is issued from one member of a sysplex data sharing group, RACF attempts to connect every peer system that is in data sharing mode or in read-only mode to the coupling facility structures associated with each active database. If any connection attempt fails, the affected member enters read-only mode, although the data set will remain allocated and active. The system originating the command might be in either data sharing mode or in read-only mode.

**INACTIVE [NOCLASSACT(class-namelist | *) (NOTAPE)]**

**INACTIVE**

Specifies that the RACF function for, and access to, the RACF database is to be deactivated.

To deactivate a particular primary data set or a backup data set, specify the DATASET operand with the appropriate data set name. If the DATASET operand is not specified, the default is all primary RACF data sets.

If your installation did not specify a backup database in the data set name table (ICHRDSNT), and you need to deactivate the primary database, you must use the RVARY INACTIVE command. If you have only a single data set, your system enters failsoft processing. If you have multiple data sets and only some are active, you are likely to experience ABENDs.

When you deactivate any RACF data set, it is automatically deallocated.

If RACF is enabled for sysplex communication, RACF disconnect from any coupling-facility-related structures that are in use by members of a RACF sysplex data sharing group running in data sharing mode or in read-only mode.

If the data set specified in the RVARY INACTIVE command is associated with a coupling facility structure failure, or with a connection failure and there is no other failed structure or connection, the system can be put into data sharing mode as a result of RVARY INACTIVE.

**INACTIVE NOCLASSACT(class-namelist | *)**

Specifies those classes for which RACF protection is not in effect while RACF is inactive. The variable *class-namelist* can contain any class defined in the class descriptor table, while * indicates that the operand applies to all classes defined in class descriptor table. This option takes effect immediately and is valid for the current IPL or until RVARY ACTIVE is issued. If you just want to deactivate the class(es) without deactivating the RACF database, you should use the command SETROPTS NOCLASSACT(class-namelist | *) since INACTIVE NOCLASSACT also deactivates the database. NOCLASSACT is not propagated when issued from a member of a sysplex data sharing group.

**INACTIVE(NOTAPE)**

Specifies that tape volume protection for volumes with IBM standard labels, ANSI labels, and nonstandard labels is no longer in effect while RACF is inactive. This option takes effect immediately and is valid for the current IPL or until RVARY ACTIVE is issued. If you just want to deactivate the tape volume without deactivating the RACF database, you should use the command SETROPTS NOCLASSACT(tapevol) since INACTIVE(NOTAPE)
RVARY

also deactivates the database. NOTAPE is not propagated when issued from a member of a sysplex data sharing group.

DATASHARE | NODATASHARE

DATASHARE
 Specifies to begin data sharing mode. This operand applies only when RACF is enabled for sysplex communication.

If the mode was non-data sharing, RACF discontinues using the hardware RESERVE/RELEASE protocol and uses GRS to serialize access to the RACF database. Systems not already in data sharing mode attempts to connect to all RACF structures. For information on failure scenarios, see [z/OS Security Server RACF System Programmer’s Guide](#).

If RACF is enabled for sysplex communication, RACF propagates RVARY DATASHARE commands to the other systems in the data sharing group. Members in non-data sharing mode attempt to connect to all structures. If successful, the mode becomes data sharing. Otherwise, it becomes read-only mode.

**Note:** The current state of the RACF system (that is, ACTIVE or FAILSOFT) from which the command is issued has no effect on either the function or the propagation of the RVARY DATASHARE commands or vice versa, unless the system is in failsoft due to an error during IPL.

NODATASHARE
 suspends data sharing mode and enables an installation to update the RACF database even if the system is experiencing coupling facility failure or unavailability. This operand applies only when RACF is enabled for sysplex communication.

When sharing data among many systems, RACF relies on the coupling facility and an alternative serialization technique to provide higher performance. In this environment, RVARY NODATASHARE might result in degraded performance, because RACF discontinues using the coupling facility cache structures and serialization associated with RACF sysplex data sharing and employs the hardware RESERVE/RELEASE protocol. It should be used only to allow critical updates to the database.

If RACF is enabled for sysplex communication, RACF propagates RVARY NODATASHARE commands to the other systems in the data sharing group. The effect of the RVARY DATASHARE command on group members depends on the member’s previous database access mode. If the member’s previous database access mode was data sharing mode or read-only mode, the member disconnects from all structures and enters non-data sharing mode. If the members previous database access mode was non-data sharing mode, no action is taken.

**Note:** The current state of the RACF system (that is, ACTIVE or FAILSOFT) from which the command is issued has no effect on either the function or the propagation of the RVARY NODATASHARE commands or vice versa, unless the system is in failsoft due to an error during IPL.

SWITCH
 Specifies that all processing is to switch from the primary RACF data sets (identified by the DATASET operand) to the corresponding backup data sets. When the switch occurs, the primary data sets are deactivated and deallocated.
If you specify DATASET(*) or omit DATASET, the command applies to all primary data sets. If you specify the name of a backup data set on the DATASET operand, RACF issues an error message and ignores the name. In order for the switch to take place, the corresponding backup data sets must be active.

When you issue RVARY SWITCH, RACF associates a set of buffers with the new primary database (the original backup database) and disassociates the buffers from the original primary database (the new backup database). The coupling facility structures associated with the primary and backup RACF databases are not switched, so IRRXCF00_Pnnn structures always correspond to primary database and IRRXCF00_Bnnn structures always correspond to backup database.

To return to the original primary database, you must first activate the backup data sets (the former primary data set) using an RVARY ACTIVE command. An RVARY SWITCH then returns the primary data sets to their original position.

If RACF is enabled for sysplex communication, RACF allocates buffers for backup data sets. The size of the buffer for the backup database is 20 percent of the primary database buffer size. When you issue RVARY SWITCH, RACF associates the larger buffer with the new primary database (the original backup database).

**Note:** If the data set specified in the SWITCH command is associated with a coupling facility structure failure or with a connection failure, and there is no other failed structure or connection, the system might be put into data sharing mode as a result of the RVARY SWITCH.

**DATASET(data-set-name ... | *)**

Specifies a list of one or more RACF data sets to be switched, reactivated, or deactivated, depending on the SWITCH, ACTIVE, or INACTIVE operands. If you specify DATASET(*) or omit DATASET, the command applies to all primary data sets.

DATASET can be specified with ACTIVE, INACTIVE, or SWITCH; it is not applicable with DATASHARE, NODATASHARE or LIST.

**Note:** As an exception to normal TSO parsing rules, RACF continues to recognize previously acceptable abbreviations (such as D, DA, DAT, DATA, DATAS) as aliases for DATASET. The shortest acceptable alias for DATASHARE is DATASH.

Do not enclose data set names in single quotation marks.

**LIST | NOLIST**

If you omit LIST and NOLIST, the default value is LIST.

**LIST**

Specifies that status information is to be listed for all RACF data sets. If you specify ACTIVE, INACTIVE, SWITCH, DATASHARE, or NODATASHARE, the status displayed is the status after the requested changes have been made if the changes were approved by the operator. If RACF is enabled for sysplex communication, the LIST output includes a line indicating the current operating mode. RVARY LIST does not require operator approval.

The volume information contains an *NA if the device on which the RACF data set resides has been dynamically reconfigured from the system. It contains *DEALLOC if the data set has been inactivated and deallocated.
If at least one RACF database volume is not shared, the SHR column is added to the volume information, and the unshared volume is marked N. The N indicates that the RACF data set resides on a device that is not shared, or it resided on a nonshared device prior to inactivation and deallocation.

**NOLIST**

Specifies that status information for RACF data sets is not to be listed.

### Examples

**Table 90. RVARY examples**

<table>
<thead>
<tr>
<th>Example</th>
<th>Operation</th>
<th>Command</th>
<th>Defaults</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>User wants to see if the backup data sets are activated.</td>
<td>RVARY LIST</td>
<td>None.</td>
</tr>
<tr>
<td>2</td>
<td>Operator wants to temporarily deactivate and deallocate RACF to make repairs to a particular primary RACF data set.</td>
<td>#RVARY INACTIVE,DATASET(RACF.PRIM1)</td>
<td>LIST</td>
</tr>
<tr>
<td>3</td>
<td>Operator wants to activate the backup data set (RACF.BACK1).</td>
<td>#RVARY ACTIVE,DATASET(RACF.BACK1)</td>
<td>LIST</td>
</tr>
<tr>
<td>4</td>
<td>Operator wants to switch from using the primary data set to using the backup data set.</td>
<td>#RVARY SWITCH,DATASET(RACF.PRIM1)</td>
<td>LIST</td>
</tr>
<tr>
<td>5</td>
<td>User wants to change the operating mode to non-data sharing mode for all members of the IRRXCF00 group, in order to allow an update of the RACF data set.</td>
<td>RVARY NODATASHARE</td>
<td>LIST</td>
</tr>
<tr>
<td>6</td>
<td>User wants to change the operating mode from non-data sharing mode to data sharing mode in order to make use of coupling facility performance enhancements.</td>
<td>RVARY DATASHARE</td>
<td>LIST</td>
</tr>
</tbody>
</table>
RVARY

<table>
<thead>
<tr>
<th>ICH15013I RACF DATABASE STATUS:</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACTIVE USE NUM VOLUME DATASET SHR</td>
</tr>
<tr>
<td>------ --- --- ------ ------- ---</td>
</tr>
<tr>
<td>YES PRIM 1 D94RF1 RACF.PRIM.R17.P1</td>
</tr>
<tr>
<td>YES BACK 1 D94RF2 RACF.BACK.R17.B1 N</td>
</tr>
<tr>
<td>YES PRIM 2 D94RF1 RACF.PRIM.R17.P2</td>
</tr>
<tr>
<td>YES BACK 2 D94RF2 RACF.BACK.R17.B2 N</td>
</tr>
</tbody>
</table>

Figure 67. Example 1: Output for the RVARY LIST command

<table>
<thead>
<tr>
<th>ICH15013I RACF DATABASE STATUS:</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACTIVE USE NUM VOLUME DATASET</td>
</tr>
<tr>
<td>------ --- --- ------ -------</td>
</tr>
<tr>
<td>NO PRIM 1 *DEALLOC RACF.PRIM1</td>
</tr>
<tr>
<td>NO BACK 1 D94RF1 RACF.BACK1</td>
</tr>
<tr>
<td>YES PRIM 2 D94RF1 RACF.PRIM2</td>
</tr>
<tr>
<td>NO BACK 2 D94RF1 RACF.BACK2</td>
</tr>
<tr>
<td>YES PRIM 3 D94RF1 RACF.PRIM3</td>
</tr>
<tr>
<td>NO BACK 3 D94RF1 RACF.BACK3</td>
</tr>
</tbody>
</table>

Figure 68. Example 2: Output following deactivation and deallocation of RACF.PRIM1

<table>
<thead>
<tr>
<th>ICH15013I RACF DATABASE STATUS:</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACTIVE USE NUM VOLUME DATASET</td>
</tr>
<tr>
<td>------ --- --- ------ -------</td>
</tr>
<tr>
<td>NO PRIM 1 *DEALLOC RACF.PRIM1</td>
</tr>
<tr>
<td>YES BACK 1 D94RF1 RACF.BACK1</td>
</tr>
<tr>
<td>YES PRIM 2 D94RF1 RACF.PRIM2</td>
</tr>
<tr>
<td>NO BACK 2 D94RF1 RACF.BACK2</td>
</tr>
<tr>
<td>YES PRIM 3 D94RF1 RACF.PRIM3</td>
</tr>
<tr>
<td>NO BACK 3 D94RF1 RACF.BACK3</td>
</tr>
</tbody>
</table>

Figure 69. Example 3: Output following the activation of RACF.BACK1

<table>
<thead>
<tr>
<th>ICH15013I RACF DATABASE STATUS:</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACTIVE USE NUM VOLUME DATASET</td>
</tr>
<tr>
<td>------ --- --- ------ -------</td>
</tr>
<tr>
<td>NO BACK 1 *DEALLOC RACF.PRIM1</td>
</tr>
<tr>
<td>YES PRIM 2 D94RF1 RACF.PRIM2</td>
</tr>
<tr>
<td>NO BACK 2 D94RF1 RACF.BACK2</td>
</tr>
<tr>
<td>YES PRIM 3 D94RF1 RACF.PRIM3</td>
</tr>
<tr>
<td>NO BACK 3 D94RF1 RACF.BACK3</td>
</tr>
</tbody>
</table>

Figure 70. Example 4: Output following the RVARY SWITCH,DATASET(RACF.PRIM1) command
RVARY

| ICH15019I Initiating propagation of RVARY command to members of RACF data sharing group IRRXCF00 |
| ICH15013I RACF DATABASE STATUS: |
| ACTIVE USE NUM VOLUME DATASET |
| ------ --- --- ------ ------- |
| YES  PRIM 1 D94RF1 RACF.BACK1 |
| NO   BACK 1 *DEALLOC RACF.PRIM1 |
| YES  PRIM 2 D94RF1 RACF.PRIM2 |
| NO   BACK 2 D94RF1 RACF.BACK2 |
| YES  PRIM 3 D94RF1 RACF.PRIM3 |
| NO   BACK 3 D94RF1 RACF.BACK3 |
| MEMBER SYS1 IS SYSPLEX COMMUNICATIONS ENABLED & IN DATA SHARING MODE. |
| ICH15020 RVARY command has finished processing. |

Figure 71. Example 5: Output following the RVARY NODATASHARE command

| ICH15019I Initiating propagation of RVARY command to members of RACF data sharing group IRRXCF00 |
| ICH15013I RACF DATABASE STATUS: |
| ACTIVE USE NUM VOLUME DATASET |
| ------ --- --- ------ ------- |
| YES  PRIM 1 D94RF1 RACF.BACK1 |
| NO   BACK 1 *DEALLOC RACF.PRIM1 |
| YES  PRIM 2 D94RF1 RACF.PRIM2 |
| NO   BACK 2 D94RF1 RACF.BACK2 |
| YES  PRIM 3 D94RF1 RACF.PRIM3 |
| NO   BACK 3 D94RF1 RACF.BACK3 |
| MEMBER SYS1 IS SYSPLEX COMMUNICATIONS ENABLED & IN DATA SHARING MODE. |
| ICH15020 RVARY command has finished processing. |

Figure 72. Example 6: Output following the RVARY DATASHARE command
SEARCH (Search RACF database)

Purpose

Use the SEARCH command to obtain a list of RACF profiles, users, and groups.
You can request one or more of the following:

- Profile names that contain a specific character string.
- Profiles for resources that have not been referenced for more than a specific
  number of days.
- Profiles that RACF recognizes as model profiles.
- Data set and general resource profiles that contain a level equal to or greater
  than the level you specify.
- User and resource profiles that contain a security label that matches the security
  label you specify.
- User and resource profiles that contain a security level that matches the security
  level that you specify.
- User and resource profiles that contain an access category that matches the
  access category that you specify.
- User profiles that contain an OMVS UID equal to the UID you specify.
- Group profiles that contain an OMVS GID equal to the GID you specify.
- Profiles for tape volumes that contain only data sets with an expiration date that
  matches the criteria you specify.
- Profiles for data sets that reside on specific volumes (or VSAM data sets that are
  cataloged in catalogs on specific volumes).
- Profiles for tape data sets, non-VSAM DASD data sets, or VSAM data sets.

You can display the selected profile names at your terminal.

You can also format the selected profile names with specific character strings into a
series of commands or messages and retain them in a CLIST data set.

Restriction: When searching profiles in the IDIDMAP class, you cannot use the
FILTER or MASK option to limit the results of the search. This is because IDIDMAP
profile names are stored in UTF-8 format and are translated to EBCDIC for use with
the SEARCH command.

RACF date handling: RACF interprets dates with 2-digit years as follows. (The yy
value represents the 2-digit year.)
- If 70 < yy <= 99, the date is interpreted as 19yy.
- If 00 <= yy <= 70, the date is interpreted as 20yy.

Issuing options

The following table identifies the eligible options for issuing the SEARCH command:

<table>
<thead>
<tr>
<th>As a RACF TSO command?</th>
<th>As a RACF operator command?</th>
<th>With command direction?</th>
<th>With automatic command direction?</th>
<th>From the RACF parameter library?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>Yes</td>
<td>Yes. (See rule.)</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Rule: The SEARCH command is not eligible for command direction when the CLIST
keyword is specified.
SEARCH

For information on issuing this command as a RACF TSO command, refer to Chapter 3, “RACF TSO commands,” on page 15.

For information on issuing this command as a RACF operator command, refer to Chapter 4, “RACF operator commands,” on page 21.

You must be logged on to the console to issue this command as a RACF operator command.

Related commands

- To obtain information on general resource profiles, see RLIST (List general resource profile).
- To display a data set profile, see LISTDSD (List data set profile).
- To display a user profile, see LISTUSER (List user profile).
- To display a group profile, see LISTGRP (List group profile).

Authorization required

When issuing the SEARCH command as a RACF operator command, you might require sufficient authority to the proper resource in the OPERCMDS class. See z/OS Security Server RACF Security Administrator’s Guide for further information.

You must have a sufficient level of authority for each profile selected as the result of your request, such that one of the following conditions is met:

- You have the SPECIAL attribute,
- You have the system-AUDITOR attribute,
- The profile is within the scope of a group in which you have either the group-SPECIAL or group-AUDITOR attribute, or

If none of the above is true, one of the following must be true:

- If the profile is for a DASD data set, the high-level qualifier of the data set name (or the qualifier supplied by a command installation exit) is your user ID.
- If the profile is in the FILE or DIRECTRY class, the second qualifier of the profile name is your user ID.
- You are on the access list for the profile and you have at least READ authority.
- Your current connect group (or, if list-of-groups checking is active, any group to which you are connected) is on the access list and has at least READ authority.
- You have the OPERATIONS attribute, or the profile is within the scope of a group in which you have the group-OPERATIONS attribute, and the class is DATASET or a general resource class that specifies OPER=YES in the static class descriptor table or OPERATIONS(YES) in the dynamic class descriptor table.
- The universal access authority is at least READ (or GLOBAL when listing discrete profiles).

Note: If the SECLABEL class is active, your current security label must dominate the security label of the DASD data set or general resource profile.

In order to use the USER operand, one of the following must be true:

- You have the SPECIAL or system-AUDITOR attribute.
- You are the owner of the specified user profile.
- You enter your own user ID on the USER operand.
SEARCH

- You have the group-SPECIAL or group-AUDITOR attribute in a group that owns the user profile.

In addition to one of the other four conditions, RACF also checks your security level and categories against those in the specified user profile.

To specify the AT keyword, you must have READ authority to the DIRECT.node resource in the RRSFDATA class and a user ID association must be established between the specified node.userid pair(s).

To specify the ONLYAT keyword you must have the SPECIAL attribute, the userid specified on the ONLYAT keyword must have the SPECIAL attribute, and a user ID association must be established between the specified node.userid pair(s) if the user IDs are not identical.

Note that it is the authority of the user ID specified on the USER operand that is used to determine if SEARCH displays the profile name.

No authorization is required to the user or group profiles that are listed when the UID or GID keyword is specified.

Inactive SECLABEL profiles and profiles that contain inactive security labels may not be listed if SETROPTS SECL LABELSYSTEM is active because only users with SPECIAL or AUDITOR authority are allowed to view inactive security labels.

### Syntax

For the key to the symbols used in the command syntax diagrams, see "Syntax of RACF commands and operands" on page 9. The complete syntax of the SEARCH command is:

```
[subsystem-prefix][SEARCH \ SR]
  [ AGE(number-of-days) ]
  [ [ ALL | GENERIC | NOGENERIC | MODEL | TAPE | VSAM | NONVSAM ] ]
  [ AT([node].userid ...) | ONLYAT([node].userid ...) ]
  [ CATEGORY([category-name] ]
    | EXPIRES(number-of-days)
    | LEVEL(level-number)
    | SECLABEL([seclabel-name] ]
    | SECLEVEL([seclevel-name] ]
    | WARNING) ]
  [ CLASS( [DATASET | class-name] ) ]
  [ CLIST (['string-1' ['string-2'] ])]
  [ FILTER(filter-string) ]
  [ GID (group-identifier) ]
  [ [LIST | NOLIST] ]
  [ [MASK([char-1] *) [char-2]] | NOMASK] ]
  [ UID (user-identifier) ]
  [ USER (userid) ]
  [ VOLUME ]
  [ VOLUME(volume-serial) ]
```

For information on issuing this command as a RACF TSO command, refer to Chapter 3, “RACF TSO commands,” on page 15.
SEARCH

For information on issuing this command as a RACF operator command, refer to Chapter 4, “RACF operator commands,” on page 21.

Parameters

$subsystem-prefix$
Specifies that the RACF subsystem is the processing environment of the command. The subsystem prefix can be either the installation-defined prefix for RACF (1–8 characters) or, if no prefix has been defined, the RACF subsystem name followed by a blank. If the command prefix was registered with CPF, you can use the MVS command D OPDATA to display it or you can contact your RACF security administrator.

Only specify the subsystem prefix when issuing this command as a RACF operator command. The subsystem prefix is required when issuing RACF operator commands.

$AGE(number-of-days)$
Specifies the aging factor to be used as part of the search criteria.

Note: This operand works only for discrete profiles and requires that STATISTICS is enabled system-wide.

Only resources that have not been referenced within the specified number of days are selected, unless you specify CLASS(GROUP). In this case, the SEARCH command uses the date on which the group was defined to determine the age.

You can specify up to five digits for $number-of-days$.

$ALL | GENERIC | NOGENERIC | MODEL | TAPE | VSAM | NONVSAM$

$ALL$
Specifies that RACF is to select all data set profiles (tape, VSAM, and non-VSAM DASD) including both generic and discrete profiles. RACF ignores this operand for classes other than DATASET. ALL is the default if you omit VSAM, NONVSAM, TAPE, GENERIC, NOGENERIC, MODEL, and ALL.

$GENERIC$
Specifies that only generic profiles are to be selected. If neither GENERIC nor NOGENERIC is specified, both profile types are selected. RACF ignores this operand unless generic profile command processing is enabled.

RACF ignores this operand unless generic profile command processing is enabled.

$NOGENERIC$
Specifies that no generic profiles (that is, only discrete profiles) are to be selected. If neither GENERIC nor NOGENERIC is specified, both profile types are selected.

RACF ignores this operand unless generic profile command processing is enabled.

$MODEL$
Specifies that only data set profiles having the MODEL attribute are to be selected. RACF ignores this operand for classes other than DATASET.

$TAPE$
Specifies that only tape data sets are to be selected. RACF ignores this operand for classes other than DATASET.
VSAM
Specifies that only VSAM data sets are to be selected. RACF ignores this operand for classes other than DATASET.

NONVSAM
Specifies that only non-VSAM data sets are to be selected. RACF ignores this operand for classes other than DATASET.

AT | ONLYAT
The AT and ONLYAT keywords are only valid when the command is issued as a RACF TSO command.

AT([node],userid ...)
Specifies that the command is to be directed to the node specified by node, where it runs under the authority of the user specified by userid in the RACF subsystem address space.

If node is not specified, the command is directed to the local node.

Note: The SEARCH command is not eligible for command direction when the CLIST keyword is specified. Do not specify the AT and CLIST keywords together on a SEARCH command.

ONLYAT([node],userid ...)
SEARCH is not eligible for automatic command direction. If you specify the ONLYAT keyword, the effect is the same as if you specified the AT keyword.

CATEGORY | EXPIRES | LEVEL | SECLEVEL | SECLABEL | WARNING

CATEGORY[(category-name)]
Specifies that RACF is to select only profiles with an access category matching the category name that you specify, where category-name is an installation-defined name that is a member of the CATEGORY profile in the SECDATA class. If you specify CATEGORY and omit category-name, RACF selects only profiles that contain undefined access category names (names that were once known to RACF but that are no longer valid).

RACF ignores this operand when CLASS(GROUP) is specified.

EXPIRES(number-of-days)
 Specifies that RACF is to select only tape volumes on which all of the data sets either have expired or will expire within the number of days that you specify. The variable number-of-days is a number of 1–5 digits in length in the range of 0–65533. For data sets that never expire, use 99999. RACF ignores this operand for classes other than TAPEVOL.

LEVEL(level-number)
Specifies that RACF is to select only profiles with an installation-defined level that equals the level number you specify. You can specify a value for level of 0–99.

RACF ignores this operand for classes other than DATASET or classes defined in the RACF class descriptor table.

SECLABEL[(seclabel-name)]
Specifies that RACF is to select only profiles with a security label name that matches the value you specify for seclabel.

SECLEVEL[(seclevel-name)]
Specifies that RACF is to select only profiles with a security level name that matches seclevel-name, where seclevel-name is an installation-defined name that is a member of the SECLEVEL profile in the SECDATA class. If
you specify SECLEVEL and omit seclen-name, RACF selects only profiles that contain undefined security level names (names that were once known to RACF but that are no longer valid).

RACF ignores this operand when you specify CLASS(GROUP).

WARNING
Specifies that only resources with the WARNING indicator are to be selected.

RACF ignores this operand when you specify CLASS as USER or GROUP.

CLASS(DATASET | class-name)
Specifies the name of the class of profiles to be searched. The valid resource classes are DATASET, USER, GROUP, and those specified in the class descriptor table. For a list of general resource classes defined in the class descriptor table supplied by IBM, see Appendix B, “Supplied RACF resource classes,” on page 659.

If you omit this operand, the default value is DATASET.

To search all RACF-defined user profiles, you must have either the SPECIAL or AUDITOR attribute.

SEARCH CLASS(USER) can be issued to obtain information about the irrcerta and irrsitec user IDs, which are the user IDs used by RACF to anchor digital certificates.

When searching with the CLASS(GROUP) option, groups are listed based upon the connect authority of the user, not READ or higher access to the profile. If CLASS(TAPEVOL) is specified, RACF processes all volumes that meet the search criteria independently, even if the volumes belong to a tape volume set.

CLIST(["string-1 [' string-2']"])  
Specifies that the selected profile names are to be retained in a CLIST data set. One record is put into the data set for each selected profile name.

Profile names containing ampersands (&) appear in the CLIST data set with each occurrence of an ampersand (&) doubled (&&). When the CLIST is executed, double ampersands (&&) prevent the CLIST from performing symbolic substitution when encountering a variable. The CLIST removes only the first ampersand, leaving the second ampersand intact.

"string-1 [' string-2']"  
Specifies strings of alphanumeric characters that are put into the CLIST records along with the selected profile names. Each string must be enclosed in single quotation marks. In this way, you can build a set of commands that are similar except for the profile name.

Mixed-case strings are always accepted and preserved for the CLIST operand. If string-1 is specified, the resulting output CLIST will contain a CONTROL ASIS statement.

The format of the text portion of the CLIST record is as follows:

string-1'data-set name' or
string-1'volume-serial-number' or
string-1'terminal-name' or
string-1

Recommendation: No blank is inserted after string-1 or before string-2. To ensure that the commands execute correctly, use a blank character as the last character in string-1 and the first character in string-2. For example, specify:

CLIST('DELDSD ' SET')
SEARCH

rather than:

CLIST('DELDSD''SET')

An 8-position sequence number is placed on the front of the text.

If both strings are missing, the CLIST record contains only the profile name.

If you want a string of data to appear only after the resource name, specify string-1 as a double- quotation mark (").

The DASD data set name for the CLIST data set is generated in the format:

'prefix.EXEC.RACF.CLIST'

where prefix is the default data set name prefix in your TSO profile. If you do not have a prefix specified in your TSO profile, (PROFILE NOPREFIX), the user ID from the SEARCH command issuer's ACEE is used as the qualifying prefix.

If this data set is partitioned rather than sequential, the CLIST records are placed in member TEMPNAME of the data set. In either case, you can execute the CLIST after SEARCH has finished by issuing the TSO/E command:

EXEC 'prefix.EXEC.RACF.CLIST'

If a CLIST data set is found through the catalog and is a sequential data set, the records it contains are replaced with the new records. If the CLIST data set is a partitioned data set, however, member TEMPNAME is created to hold the new records, or is replaced if the member already exists.

If the CLIST data set does not already exist, it is created and cataloged. If the CLIST data set created is a partitioned data set, member TEMPNAME is created.

The CLIST data set must have variable length records and a maximum logical record size of 255. This includes a 4-byte length field at the front of the record. The records are numbered in sequence by 10.

Note: The SEARCH command is not eligible for command direction when the CLIST keyword is specified. Do not specify the AT and CLIST keywords together on a SEARCH command.

FILTER(filter-string)

(Also see the MASK operand.)

Specifies the string of alphanumeric characters used to search the RACF database. The filter string defines the range of profile names you wish to select from the RACF database. For a tape or DASD data set name, the filter string length must not exceed 44 characters. For a general resource class, the filter string length must not exceed the length of the profile name specified in the class descriptor table.

Mixed-case strings are accepted and preserved when CLASS refers to a class defined in the static class descriptor table with CASE=ASIS or in the dynamic class descriptor table with CASE(ASIS).

When you issue the SEARCH command with the FILTER operand, RACF lists profile names from the RACF database matching the search criteria specified in the filter string. Note that RACF lists only those profile names that you are authorized to see.

The following generic characters have special meaning when used as part of the filter string:
You can use the percent sign to represent any one character in the profile name, including a generic character. For example, if you specify DASD%% as a filter string, it can represent profile names such as DASD01, DASD2A, and DASD5. If you specify %%%% as a filter string, it can represent profile names DASD1, DASD2, DASD%%%, TAPE%, MY%%%, TAPE*, and %%%*.

You can use a single asterisk to represent zero or more characters in a qualifier, including generic characters. For example, AB*.CD can represent data set profile names such as AB.CD, ABEF.CD, and ABX.CD. ABC.D* can represent data set profile names such as ABC.DEFG, ABC.D%%%%, and ABC.D%%*. If you specify a single asterisk as the only character in a qualifier, it represents the entire qualifier. For example, ABC.* represents data set profile names such as ABC.D, ABC.DEF, ABC.mmm%, and ABC.%DE.

For general resource and data set profile names, you can use a double asterisk to represent zero or more qualifiers in the profile name. For example, AB.**.CD represents data set profile names such as AB.CD, AB.DE.EF.CD, and AB.XYZ.CD. You cannot specify other characters with ** within a qualifier. (For example, you can specify FILTER(USER1.A**), but not FILTER(USER1.A**). You can also specify ** as the only characters in the filter-string to represent any entire profile name.

Tip: Use FILTER for an alternative to MASK | NOMASK as a method for searching the RACF database. FILTER offers more flexibility than MASK. For example, when you use FILTER, you can generalize the character string you specify to match multiple qualifiers or multiple characters within a profile name. You can also specify a character string to match a single character regardless of its value or search for a character string anywhere in a profile name.

Restrictions:
- The SEARCH command might provide unpredictable results when searching on the DIGTCERT or DIGTRING classes. Because these classes contain names with mixed-case characters, the profile filter on the SEARCH command might not function correctly.
- You cannot use a generic character (+, **, or %) in the high-level qualifier when you define a generic profile for a data set. However, you can use a generic character in the high-level qualifier of a data set name when specifying a filter-string with the FILTER operand.
- The FILTER and MASK | NOMASK operands are mutually exclusive; you cannot specify FILTER with either MASK or NOMASK on the same SEARCH command.
- When searching profiles in the IDIDMAP class, you cannot use FILTER to limit the results of the search. This is because IDIDMAP profile names are stored in UTF-8 format and are translated to EBCDIC for use with the SEARCH command.

GID (group-identifier)
Specifies that RACF is to display all group profiles which contain the specified group-identifier for the GID in the OMVS segment. GID is ignored unless CLASS(GROUP) is specified. When GID is specified, all other keywords (except CLASS) are ignored.

LIST | NOLIST
LIST
Specifies that the selected data set names, volume serial numbers, or terminal names are to be displayed at your terminal. LIST is the default value when you omit both LIST and NOLIST.
SEARCH

NOLIST
Specifies that the selected data set names, volume serial numbers, or terminal names are not to be displayed at your terminal. You can use this operand only when you specify the CLIST operand. If you use NOLIST without CLIST, the command fails.

MASK | NOMASK

MASK(char-1 | * [char-2])
(Also see the FILTER operand.)
Specifies the strings of alphanumeric characters used to search the RACF database. This data defines the range of profile names selected. The two character strings together must not exceed 44 characters for a tape or DASD data set name, or, for general resource classes, the length specified in the class descriptor table.

char-1
Specifies the starting characters of names of profiles to be searched. The string can be any length up to the maximum allowable length of the resource name. All profiles that start with char-1 in their resource names are selected.

If an asterisk (*) is specified for char-1, it specifies that profiles of the search criteria are to be selected:
• For DATASET class, your user ID is used as the mask for the profiles to be selected.
• For other classes, all profiles of the specified class are selected.

char-2
Specifies a second string of characters to be included in the search for profiles. All profiles whose names start with char-1 and contain char-2 anywhere beyond char-1 are selected. This limits the list to a subset of the resource names identified with char-1.

If an asterisk (*) is specified instead of char-1, all profiles that contain char-2 anywhere in their resource names are selected.

If you omit both the MASK and NOMASK operands, this is the same as specifying MASK(*): for the DATASET class, your user ID is used as the mask for profiles to be selected; for other classes, all profiles of the class are selected. (Note also that for classes other than DATASET, omitting both operands is the same as NOMASK.)

Mixed-case strings are accepted and preserved when CLASS refers to a class defined in the static class descriptor table with CASE=ASIS or in the dynamic class descriptor table with CASE(ASIS).

Restriction: When searching profiles in the IDIDMAP class, you cannot use MASK to limit the results of the search. This is because IDIDMAP profile names are stored in UTF-8 format and are translated to EBCDIC for use with the SEARCH command.

NOMASK
Specifies that RACF is to select all profiles (to which you are authorized) in the specified class.

Note: The MASK | NOMASK and FILTER operands are mutually exclusive. You cannot specify MASK or NOMASK with FILTER on the same SEARCH command.
**SEARCH**

**UID(user-identifier)**
Specifies that RACF is to display all user profiles which contain the specified user-identifier for the UID in the OMVS segment. UID is ignored unless CLASS(USER) is specified. When UID is specified, all other keywords (except CLASS) are ignored.

**USER(userid)**
Specifies that RACF is to list the profiles that the specified user has access to (READ authority or higher, or owner) for the class you specify on the CLASS operand. RACF lists only those profiles that the specified owner is allowed to see.

If you issue:
SEARCH USER(JONES) CLASS(ACCTNUM)

RACF lists all TSO account numbers that user ID JONES is allowed to use.

If you issue:
SEARCH USER(JONES) NOMASK

RACF lists profiles in the DATASET class that JONES has access to.

If you issue:
SEARCH USER(JONES) CLASS(GROUP)

RACF lists all groups that user ID JONES owns or, in which JONES has JOIN or CONNECT authority or the group-SPECIAL attribute.

**Notes:**
1. If you omit the CLASS operand, the default class is DATASET. For more information, see the description of the CLASS operand.
2. You should not specify a user ID that has been revoked. If you need to display information about a user whose user ID is revoked, perform the following steps:
   a. Change the password for the user ID.
   b. Resume the user ID.
   c. Issue the SEARCH command to display the desired information.
   d. Revoke the user ID.
3. You can only specify one user ID at a time on the USER operand. If you need to display information about all users, first create a CLIST by issuing the following command:
   SEARCH CLASS(USER) CLIST('SEARCH USER(' ') CLASS(class-name)')
   After you create a CLIST, issue:
   EXEC 'prefix.EXEC.RACF.CLIST'
   to display the desired information. (Note that prefix is the default data set name prefix in your TSO profile.) For more information, see the description of the CLIST operand.

**VOLUME**
Specifies that you want RACF to display volume information for each tape or DASD data set that meets the search criteria specified by the MASK or FILTER operand.

RACF ignores this operand if you specify GENERIC.
For non-VSAM data sets, the volume serial number displayed is the location of the data set. For VSAM data sets, the volume serial number displayed is the location of the catalog entry for the data set. For tape data sets, the volume serial number displayed is the location of the TVTOC entry for the data set.

This operand is valid only for CLASS(DATASET). RACF ignores it for all other class values.

**VOLUME(volume-serial ...)**

Specifies the volumes to be searched; the volume serial numbers become part of the search criteria. Non-VSAM DASD data sets are selected if they reside on the specified volumes. VSAM data sets are selected if the catalog entries for the data sets reside on the specified volumes. Tape data sets are selected if the TVTOC entries for the data set reside on the specified volumes.

RACF ignores this operand if you specify GENERIC.

If the selected data set names are displayed at your terminal, the volume information is included with each data set name.

This operand is valid only for CLASS(DATASET). RACF ignores it for all other class values.

**Examples**

*Table 91. SEARCH examples*

**Example 1**

*Operation* User CD0 wants to list all of her RACF data set profiles.

*Known* User CD0 is RACF-defined. User CD0 wants to issue the command as a RACF TSO command.

*Command* SEARCH

*Defaults* MASK(CD0) CLASS(DATASET) LIST ALL

*Results* A list of all profiles in the DATASET class beginning with CD0.

**Example 2**

*Operation* User IA0 wants to remove the RACF profiles for all DATA-type data sets for the group RESEARCH that have not been referenced for 90 days. The user wants a CLIST data set to be created with DELDSD commands for each profile satisfying the search criteria. A list is not desired.

*Known* User IA0 is connected to group RESEARCH (and is the owner of all profiles in group RESEARCH) with the group-SPECIAL attribute. User IA0 wants to issue the command as a RACF TSO command.

*Command* SEARCH FILTER(RESEARCH.DATA) AGE(90) CLIST('DELDSD ') NOLIST or

SEARCH MASK(RESEARCH.DATA) AGE(90) CLIST('DELDSD ') NOLIST

*Defaults* CLASS(DATASET) ALL

*Results* A CLIST data set with the name IA0.EXEC.RACF.CLIST is built, and the records in it are in the format:

DELDSD 'data-set-name'

**Example 3**

*Operation* User ADMIN wants to obtain a list of all data set profiles, both discrete and generic, that have the word DATA as the second-level qualifier.

*Known* User ADMIN has the SPECIAL attribute. User ADMIN wants to issue the command as a RACF operator command, and the RACF subsystem prefix is @.

*Command* @SEARCH FILTER(*.DATA.**)

*Defaults* CLASS(DATASET) LIST ALL

*Results* A list of all profiles in the DATASET class with the word DATA as the second-level qualifier. For example, the list might include data sets with names such as RESEARCH.DATA, TEST.DATA, USER.DATA.WEEK1, or GROUP.DATA.TEST.ONE.
Table 91. SEARCH examples (continued)

Example 4

**Operation** User ADM1 wants to obtain a list of all data set profiles, both discrete and generic, having a qualifier (any level) that begins with the word TEST and contains only one additional character (such as TEST1, TEST2, or TESTA).

**Known** User ADM1 has the SPECIAL attribute. User ADM1 wants to issue the command as a RACF TSO command.

**Command** SEARCH FILTER(**.TEST%.**)  
**Defaults** CLASS(DATASET) LIST ALL  
**Results** A list of all profiles in the DATASET class having a qualifier of any level that begins with the word TEST and contains only one additional character. For example, the list might include data sets with names such as RESEARCH.TEST1, TEST2.DAT, MY.TEST4.DAT, MY.TEST%.%, USER.DATA.TEST5, USER.DATA.TEST%.%, or GROUP.DATA.TESTC.FUN.

Example 5

**Operation** User ADMIN wishes to find and revoke all user IDs of users who have not accessed the system in the last 90 days. For this to work, the INITSTATS option (specified on the SETROPTS command) must be in effect.

**Known** User ADMIN has the SPECIAL attribute. User ADMIN wants to issue the command as a RACF TSO command.

**Command** SEARCH CLASS(USER) AGE(90) CLIST('ALTUSER ' 'REVOKE')  
**Defaults** Process all user ID entries.  
**Results** A CLIST data set with the name ADMIN.EXEC.RACF.CLIST listing the user ID for each user that has not accessed the system within 90 days, with records in the following format:

```
ALTUSER userid REVOKE
```

Example 6

**Operation** User ADM1 wants to get a list of all generic profiles for group SALES.

**Known** User ADM1 has the SPECIAL attribute. User ADM1 wants to issue the command as a RACF TSO command.

**Command** SEARCH MASK(SALES.*)  
**Defaults** CLASS(DATASET) LIST ALL  
**Results** A list of all profiles in the DATASET class beginning with SALES.*. (Since the string specified contains an asterisk, this list consists only of generic profiles.)

Example 7

**Operation** User ADM1 wants to get a list of all data set profiles that include a security level of CONFIDENTIAL. User ADM1 wants to direct the command to run at the local node under the authority of user HICKS.

**Known** User HICKS has the SPECIAL attribute. The CONFIDENTIAL security level has been defined to RACF. User ADM1 wants to issue the command as a RACF TSO command. Users ADM1 and HICKS have an already established user ID association.

**Command** SEARCH CLASS(DATASET) SECLEVEL(CONFIDENTIAL) AT(.HICKS)  
**Defaults** LIST ALL  
**Results** A list of all profiles in the DATASET class with a security level of CONFIDENTIAL.
SET

Purpose

Use the SET command to:

- List information related to RRSF on the local node
- Specify the name of a member of the RACF parameter library to be processed by RACF
- Set tracing on or off for specified RACF subsystem facilities
- Enable and specify options for automatic direction

Note: You might find it useful to fill out the “Configuration Worksheet” in the z/OS Security Server RACF System Programmer’s Guide to help you determine the information you need to issue the SET command.

Issuing options

The following table identifies the eligible options for issuing the SET command:

<table>
<thead>
<tr>
<th>As a RACF TSO command?</th>
<th>As a RACF operator command?</th>
<th>With command direction?</th>
<th>With automatic command direction?</th>
<th>From the RACF parameter library?</th>
</tr>
</thead>
<tbody>
<tr>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>

For information on issuing this command as a RACF operator command, see “Rules for entering RACF operator commands” on page 21. You must be logged on to the console to issue this command as a RACF operator command.

Related commands

To define an RRSF node, see TARGET (Define RRSF nodes)” on page 633.

Authorization required

When issuing the SET command as a RACF operator command, you might require sufficient authority to the proper resource in the OPERCMDS class. See z/OS Security Server RACF Security Administrator’s Guide for further information.

Syntax

For the key to the symbols used in the command syntax diagrams, see Syntax of RACF commands and operands” on page 9. The of the SET command is:
For information on issuing this command as a RACF operator command, see "Rules for entering RACF operator commands" on page 21. You must be logged on to the console to issue this command as a RACF operator command.

Parameters

**subsystem-prefix**

Specifies that the RACF subsystem is the processing environment of the command. The subsystem prefix can be either the installation-defined prefix for RACF (1–8 characters) or, if no prefix has been defined, the RACF subsystem name followed by a blank. If the command prefix was registered with CPF, you can use the MVS command D OPDATA to display it or you can contact your RACF security administrator.

Only specify the subsystem prefix when issuing this command as a RACF operator command. The subsystem prefix is required when issuing RACF operator commands.
Most settings specified by the SET command stay in effect only while the RACF subsystem address space is active. If the RACF subsystem address space is stopped, these settings are not saved. When the RACF subsystem address space is restarted, default initial values are in effect until overwritten by the new SET commands for all values except RACROUTE, CALLABLE, PDCALLABLE, DATABASE, ASID and JOBNAME. These values are saved and remain in effect after the subsystem is restarted.

AUTOAPPL | NOAUTOAPPL

**AUTOAPPL**

Specifies that automatic direction of application updates is to be activated. Profiles in the RRSFDATA class control which application updates get automatically directed to which remote nodes. See [z/OS Security Server RACF Security Administrator's Guide](z/OS Security Server RACF Security Administrator's Guide) for more information on using the RRSFDATA class to control automatic direction of application updates and for planning information that is necessary before using it.

The operands on the AUTOAPPL keyword specify who gets the result and output from automatically directed application updates.

**NOAUTOAPPL**

Specifies that automatic direction of application updates is to be deactivated. This option prevents application update requests from being directed to remote nodes.

The initial value is NOAUTOAPPL.

When SET NOAUTOAPPL is issued, the settings of OUTPUT and NOTIFY are saved. Subsequently, if a SET AUTOAPPL command is issued with no other operands, those settings are restored.

AUTODIRECT | NOAUTODIRECT

**AUTODIRECT**

Specifies that automatic command direction is to be activated. Profiles in the RRSFDATA class control which commands and password synchronization requests get automatically directed to which remote nodes. See [z/OS Security Server RACF Security Administrator's Guide](z/OS Security Server RACF Security Administrator's Guide) for more information on using the RRSFDATA class to control automatic direction and for planning information that is necessary before using automatic command direction.

The operands on the AUTODIRECT keyword specify who gets the results and output from automatically directed commands.

When SET AUTODIRECT is issued, the settings of OUTPUT and NOTIFY are saved. Subsequently, if a SET AUTODIRECT command is issued with no other operands, those settings are restored.

If you want to issue the SET AUTODIRECT command to activate both the NOTIFY and OUTPUT settings, both NOTIFY and OUTPUT must be specified on the same command for both to be in effect. If specified on two separate commands, the settings on the first invocation are lost when the second is issued. For example, if you issue SET AUTODIRECT OUTPUT and then enter SET AUTODIRECT NOTIFY, the OUTPUT setting is lost when the second command is processed. Note that AUTODIRECT NOTIFY and OUTPUT settings are independent of the AUTOAPPL, AUTOPWD, and PWSYNC settings. See the following examples for more information.
When the OUTPUT, NOOUTPUT, NOTIFY, or NONOTIFY keyword is specified, the previous values of all of these keywords is overwritten. For example, if the previous setting was:

```
OUTPUT(FAIL(NODEA.ANDREW)) NOTIFY(FAIL(NODEA.ANDREW))
```

and you wanted to also have the command issuer receive FAIL output and command results, you must use:

```
OUTPUT(FAIL(NODEA.ANDREW &RACUID)) NOTIFY(FAIL(NODEA.ANDREW &RACUID))
```

If you just specified:

```
OUTPUT(FAIL(&RACUID))
```

then NODEA.ANDREW would be removed from OUTPUT and
NODEA.ANDREW would lose his NOTIFY(FAIL) setting and the value
would return to NONOTIFY. Note, however, that these settings for
AUTOAPPL, AUTODIRECT, AUTOPWD, and PWSYNC are independent of
each other. Continuing the previous example, consider the subsequent
settings for SET AUTOPWD:

```
OUTPUT(FAIL(NODEA.ANDREW)) NOTIFY(FAIL(NODEA.ANDREW))
```

This resets the table for AUTOPWD, but leaves the previously specified
AUTODIRECT table with its &RACUID intact. AUTODIRECT, AUTOPWD, and
PWSYNC are all independent in this way, with regard to the OUTPUT and
NOTIFY settings and also about the request's processing.

**NOTIFY(notify-level(list-of-notify-users))**

Specifies that the user is to be notified (through TSO SEND command)
of the results of this RRSF request. The information sent indicates
whether the command was successful or unsuccessful, but does not
include other details about the request's processing.

**ALWAYS**

Specifies that results or output from all requests of this RRSF
function are to be returned to specified users. This option should be
used if the users are interested in the results of every request.
Output includes informational, warning, and error messages.

**WARN**

Specifies that, in the case of AUTODIRECT, results or output from
automatically directed commands are to be returned to the specified
users only when the return code from the command is 4 or greater.
In the case of AUTOAPPL, AUTOPWD, and PWSYNC, WARN is
equivalent to FAIL.

**FAIL**

Specifies that, in the case of AUTODIRECT, results or output from
automatically directed commands are to be returned to the specified
users only when the return code from the command is 8 or greater.
For AUTOAPPL, AUTOPWD, and PWSYNC, results or output from
the request are to be returned to the specified users whenever the
return code from the request is non-zero.

The list-of-notify-users value specifies up to four users who are to
receive output and/or notification of results. A user can be specified in
one of the following ways:

```
node.userid     The node and user ID separated by a period.
```
.userid

The name of the user ID on the local node preceded by a period.

&RACUID

Original issuer with regard to node and user.

When &RACUID is specified, where the results and output are sent depend on the situation. Consider the following scenarios:

- For password synchronization, automatic password direction, and automatic direction of application updates, results and output go to the specific system. For automatic command direction, the results and output go to the MAIN system. For a multisystem node, the MAIN system might not necessarily be the specific system of that issuing node.

- A user on node A directs a command to node B which results in automatic command direction to node C. &RACUID specified on node C for AUTODIRECT NOTIFY or OUTPUT sends data to node A.

- A user on node A directs a command to node B which results in a password change. This password change is propagated by password synchronization to node C. &RACUID specified on node C for PWSYNC NOTIFY or OUTPUT sends data to node B.

- A user on node A changes a password, such that automatic password direction updates the corresponding user ID on node B. This user ID propagates the password change to a peer association on that same node B. &RACUID specified on node B for PWSYNC NOTIFY or OUTPUT would send data to the original user ID on node B (not A).

**Note:** If &RACUID is specified along with the user ID from which you are issuing the command, password change (covered by password synchronization or automatic password direction), or application update, that user ID receives the output or notification twice.

If you plan to use &RACUID for application updates, be aware of the following:

- **Guideline:** Do not use &RACUID for AUTOAPPL output or notification. This might allow application updates to be done for undefined users, revoked user IDs, or the user ID of the RACF address space, and produce unexpected results during output delivery, including lost output and error messages on the console.

- Ensure all possible user ID destinations through &RACUID have the authority to create data sets. For example, an installation would not want to use &RACUID for AUTOPWD or PWSYNC if the original issuer of a password change could be a CICS user, who is unlikely to have authority to create the RRSFLIST output data set.

- If the ACEE keyword is used on the RACROUTE request, the output goes to the user ID associated with the ACEE keyword, not to the user ID of the task or address space that issued the request.
The SET command does not perform existence checking for either the user ID or node.

The combination of users specified in the list of notify users variables can be up to a maximum of four different users. In other words, the cumulative total of unique users cannot exceed 4 in both the OUTPUT and NOTIFY keywords. The same four users can be specified in each list; however, if four users are specified on one of the keywords, a fifth user cannot be specified on the other keyword. For example, if four users are specified on the OUTPUT keyword, a fifth user cannot be specified on the NOTIFY keyword.

**NONOTIFY**
Specifies that no TSO SEND commands are issued with the results of the RRSF request.

The initial value is NONOTIFY.

The allowed values for notify-level are:

**OUTPUT(output-level(list-of-output-users))**
Specifies that the output from the RRSF request should be put in the RRSFLIST data set for the user named on this keyword. If the output cannot be put in the RRSFLIST data set for any reason, the output is transmitted to the user.

Because LIST-type commands are ineligible for automatic command direction, the output usually contains messages issued during command processing, such as informational, warning, or error messages.

The valid values for output-level are the same as those described for notify-level with the NOTIFY keyword.

The valid values for list-of-output-users are the same as those described for list-of-notify-users with the NOTIFY keyword.

**NOOUTPUT**
Specifies that no output, warning, or error messages are kept or sent to anyone.

The initial value is NOOUTPUT.

**NOAUTODIRECT**
Specifies that automatic command direction is to be deactivated.

This option prevents commands from being automatically directed to remote nodes.

The initial value is NOAUTODIREC

When SET NOAUTODIRECT is issued, the settings of OUTPUT and NOTIFY are saved. Subsequently, if a SET AUTODIRECT command is issued with no other operands, those settings are restored.

**AUTOPWD | NOAUTOPWD**

**AUTOPWD**
Specifies that automatic password direction is to be activated. Profiles in the RRSFDATA class AUTODIRECT.nodename.USER.PWSYNC control which automatic password directed requests get directed to which remote nodes.

information on using the RRSFDATA class to control automatic password
direction and for planning information that is necessary before using
automatic password direction.

The operands on the AUTOPWD keyword specify who gets the result and
output from automatically directed passwords. Refer to the descriptions of
OUTPUT and NOTIFY under the AUTODIRECT keyword.

NOAUTOPWD
Specifies that automatic password direction is to be deactivated.

This option prevents passwords from being automatically directed to remote
nodes.

The initial value is NOAUTOPWD.

When SET NOAUTOPWD is issued, the settings of OUTPUT and NOTIFY
are saved. Subsequently, if a SET AUTOPWD command is issued with no
other operands, those settings are restored.

INCLUDE(member-suffix ...)
Provides the ability to specify that the contents of one or more members of the
RACF parameter library are to be processed. The INCLUDE keyword provides
a convenient mechanism to process a previously defined set of RACF
commands, such as SET and TARGET.

One or more suffixes can be specified with the INCLUDE keyword. Each
specified suffix must be:
• 1–2 characters in length
• In the alphanumeric character set (A–Z, 0–9, # (X'7B'), $ (X'5B') or @ (X'7C'))
such that when the suffix is appended to IRROPT it results in the name of a
member of the RACF parameter library.

SET INCLUDE commands can be nested in members of the RACF parameter
library. For example, if member IRROPT01 contains a SET INCLUDE
IRROPT02 command, then member IRROPT02 can contain a SET INCLUDE
IRROPT03 command.

Be careful to adhere to a hierarchical order when nesting SET INCLUDE
commands. For example, if IRROPT01 contains a SET INCLUDE IRROPT02
command, then IRROPT01 cannot contain a SET INCLUDE IRROPT01
command. Also, if IRROPT02 contains a SET INCLUDE command for any other
parameter library members, those members cannot contain a SET INCLUDE
IRROPT01 command. This restriction exists to prevent a never-ending loop of
inclusion.

If a suffix appended to IRROPT does not result in the name of a member of the
RACF parameter library, a message is issued and that suffix is ignored.

If the INCLUDE keyword is specified with any other SET keywords, the included
members are processed first. The values specified for the other keywords
override any values specified for the keywords in the included members. For
example, the values specified for TRACE override any trace values specified in
the included members.

No authorization checking or auditing is done for the commands in included
members.

JESNODE(nodename)
Specifies the name of the node needed by RRSF in the cases where returned
output from the directed commands must be transmitted to the user. RRSF
queries the primary JES system during initialization in an attempt to obtain this name automatically. This keyword should be used in the cases where RRSF cannot automatically obtain this name.

No validity checking is done on the value specified with the JESNODE keyword.

LIST
Lists the attributes of an RRSF node and trace options.

The LIST keyword provides the ability to obtain information about the RRSF node's configuration, status related to the RACF subsystem, and status of the trace options set for ASID, CALLABLE, DATABASE, JOBNAME, PDCALLABLE, and RACROUTE trace keywords.

The LIST keyword can be specified alone or in combination with other SET command keywords. When used in combination with other SET command keywords, the information displayed reflects the results after processing the other keywords.

The **rrrrrr.aaaaaaaa** information for the template version that the system is running with will be displayed in the output. For details, see the [RACF database initialization utility program (IRRMIN00)](https://www.ibm.com/support/docview.wss?uid=swg21281195) topic of [z/OS Security Server RACF System Programmer's Guide](https://www.ibm.com/support/docview.wss?uid=swg21281195).

**Note:** LIST is the default, if the SET command is issued with no keywords.

**PWSYNC | NOPWSYNC**

**PWSYNC**
Specifies that password synchronization is to be activated. See [z/OS Security Server RACF Security Administrator's Guide](https://www.ibm.com/support/docview.wss?uid=swg21281195) for more information on using the RRSFDATA class to control password synchronization and for planning information that is necessary before using password synchronization. For information on how to establish password synchronization between user IDs review the RACTION section of this manual.

The operands on the PWSYNC keyword specify who gets the result and output from password changes covered by password synchronization. Refer to the descriptions of OUTPUT and NOTIFY under the AUTODIRECT keyword.

**NOPWSYNC**
Specifies that synchronized password processing is to be deactivated.

The initial value is NOPWSYNC.

When SET NOPWSYNC is issued, the settings of OUTPUT and NOTIFY are saved. Subsequently, if a SET PWSYNC command is issued with no other operands, those settings are restored.

**TRACE**
Specifies whether tracing is to take place for the following events, using the generalized trace facility (GTF).

If the TRACE keyword is specified, at least one subkeyword must be specified to indicate whether or not tracing is to be turned on or off for each of these events. There are no defaults for these event types. For example, if APPC is the only operand specified, then the current setting for tracing IMAGE events is not changed. If IMAGE tracing was in effect, it remains in effect. Likewise, if NOIMAGE had been in effect, it would remain in effect.
The initial value is NOAPPC, NOASID, NOCALLABLE, NODATABASE, NOIMAGE, NOJOBNAME, and NORACROUTE.

The SET LIST command should always be used to verify the trace parameters have been set as expected.

The trace records are intended for use in consultation with the IBM support center when diagnosing potential RACF subsystem problems. For more information, see \textit{z/OS Security Server RACF Diagnosis Guide}.

\textbf{Important:} Trace records might contain passwords and therefore trace output data sets should be appropriately protected.

\textbf{APPC | NOAPPC}

\textbf{APPC} Specifies that tracing is to be in effect for APPC events. The trace information contains the APPC transaction return code and reason code.

\textbf{NOAPPC} Specifies that tracing is not to be in effect for APPC events.

\textbf{ASID(asid ...)} | NOASID | ALLASIDS

\textbf{ASID(asid ...)} Specify ASID to establish that you wish to trace the included list of address spaces for the CALLABLE, DATABASE, and RACROUTE keywords. When ASID(asid ...) is specified, by consecutive invocations of the SET command, the asid list is deleted and rebuilt with the current specification each time the command is issued.

\textbf{NOASID} Specify NOASID to eliminate the list of addresses to enable tracing all address spaces.

\textbf{ALLASIDS} Specifies that tracing is in effect for all address spaces.

\textbf{DATABASE | NODATABASE}

\textbf{DATABASE} Use to trace RACF database manager requests.

\textbf{ALL | NONE}

\textbf{ALL} Use ALL to enable tracing of all requests.

\textbf{NONE} Use NONE to disable tracing.

\textbf{ALTER | NOALTER}

Use ALTER to enable tracing all RACF database manager calls that change the database. Calls included are RENAMEs, ALTERs, ADDs and DELETEs. No further granularity is provided for calls that change the database.

Use NOALTER to disable tracing of all RENAMEs, ALTERs, ADDs and DELETEs.

\textbf{ALTERI | NOALTERI}

Use ALTERI to enable tracing all RACF database manager calls that change fields in the database that use ALTERI as the request.

Use NOALTERI prevent the tracing of these requests.

\textbf{READ | NOREAD}

Use READ to enable tracing all RACF database manager RACF calls that locate profiles in the database.

Use NOREAD prevent the tracing of these requests.
NODATABASE
Use NODATABASE to disable tracing database manager requests;
equivalent to DATABASE(NONE).

CALLABLE | NOCALLABLE

CALLABLE
Use to trace z/OS UNIX System Services calls.

ALL | NONE | TYPE(type ...)
Use to control the degree of tracing.

ALL  Use to enable tracing of all z/OS UNIX calls.
NONE Use to reset tracing.

TYPE(type ...)  Use to enable tracing of one or more specific z/OS UNIX calls. The TYPE operand is cumulative; issuing NOCALLABLE or CALLABLE(NONE) will reset the trace.

The request types that are supported are listed in Table 92.

Table 92. Callable services and associated function numbers

<table>
<thead>
<tr>
<th>Callable service</th>
<th>Function number</th>
<th>Callable service</th>
<th>Function number</th>
</tr>
</thead>
<tbody>
<tr>
<td>IRRSIU00</td>
<td>1</td>
<td>IRRSFK00</td>
<td>28</td>
</tr>
<tr>
<td>IRRSDU00</td>
<td>2</td>
<td>IRRSMI00</td>
<td>29</td>
</tr>
<tr>
<td>IRRSMF00</td>
<td>3</td>
<td>IRRSKI00</td>
<td>30</td>
</tr>
<tr>
<td>Reserved.</td>
<td>4</td>
<td>IRRSC100</td>
<td>31</td>
</tr>
<tr>
<td>IRRSMM00</td>
<td>5</td>
<td>IRRSC200</td>
<td>32</td>
</tr>
<tr>
<td>IRRSKA00</td>
<td>6</td>
<td>IRRSGE00</td>
<td>33</td>
</tr>
<tr>
<td>IRRSKP00</td>
<td>7</td>
<td>IRRSD100</td>
<td>34</td>
</tr>
<tr>
<td>IRRSUM00</td>
<td>8</td>
<td>IRRSDK00</td>
<td>35</td>
</tr>
<tr>
<td>IRRSGM00</td>
<td>9</td>
<td>IRRSU00</td>
<td>36</td>
</tr>
<tr>
<td>IRRSGG00</td>
<td>10</td>
<td>IRRSDA00</td>
<td>37</td>
</tr>
<tr>
<td>IRRSSU00</td>
<td>11</td>
<td>IRRSIA00</td>
<td>38</td>
</tr>
<tr>
<td>IRRSEU00</td>
<td>12</td>
<td>IRRSEQ00 (See Note.)</td>
<td>39</td>
</tr>
<tr>
<td>IRRSSG00</td>
<td>13</td>
<td>IRRSIM00</td>
<td>40</td>
</tr>
<tr>
<td>IRRSEG00</td>
<td>14</td>
<td>IRRSDL00</td>
<td>41</td>
</tr>
<tr>
<td>IRRSC000</td>
<td>15</td>
<td>IRRSMK00</td>
<td>42</td>
</tr>
<tr>
<td>IRRSCF00</td>
<td>16</td>
<td>IRRSPK00</td>
<td>43</td>
</tr>
<tr>
<td>IRRSCA00</td>
<td>17</td>
<td>IRRSPX00</td>
<td>44</td>
</tr>
<tr>
<td>IRRSEX00</td>
<td>18</td>
<td>IRRSCH00</td>
<td>45</td>
</tr>
<tr>
<td>IRRSAU00</td>
<td>19</td>
<td>IRRSPY00</td>
<td>46</td>
</tr>
<tr>
<td>IRRSK000</td>
<td>20</td>
<td>IRRSCL00</td>
<td>47</td>
</tr>
<tr>
<td>IRRSQS00</td>
<td>21</td>
<td>IRRSSB00</td>
<td>48</td>
</tr>
<tr>
<td>IRRSQF00</td>
<td>22</td>
<td>IRRSPW00</td>
<td>49</td>
</tr>
<tr>
<td>IRRSCS00</td>
<td>23</td>
<td>IRRSGS00</td>
<td>50</td>
</tr>
<tr>
<td>IRRSKF00</td>
<td>24</td>
<td>IRRSAX00</td>
<td>51</td>
</tr>
<tr>
<td>IRRSMR00</td>
<td>25</td>
<td>IRRSG100</td>
<td>52</td>
</tr>
</tbody>
</table>
### Callable services and associated function numbers (continued)

<table>
<thead>
<tr>
<th>Callable service</th>
<th>Function number</th>
<th>Callable service</th>
<th>Function number</th>
</tr>
</thead>
<tbody>
<tr>
<td>IRRSPT00</td>
<td>26</td>
<td>IRRSPS00</td>
<td>53</td>
</tr>
<tr>
<td>IRRSUG00</td>
<td>27</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Note:** Callable service IRRSEQ00 (R_Admin) has its own trace facility. For more information, see z/OS Security Server RACF Diagnosis Guide.

#### NOCALLABLE
Use NOCALLABLE to reset the trace; equivalent to CALLABLE(NONE).

#### IMAGE | NOIMAGE

**IMAGE**
- Specifies that tracing is to be in effect for IMAGE events.
- The trace information contains the command image being processed.

**NOIMAGE**
- Specifies that tracing is not to be in effect for IMAGE events.

#### JOBNAME | ALLJOBNAMES | NOJOBNAME

**JOBNAME(jobname ...)**
- Use JOBNAME to establish that you wish to trace the included list of jobs for CALLABLE, DATABASE, and RACROUTE traces. The jobname variable is a list of one or more jobs to be traced. The jobname variable will accept * at the end of jobname so as to allow matching on a set of job names. For example, if z/OS UNIX created job names of MAPES, MAPES2 and MAPES3, tracing would be facilitated through the use of * at the end of the jobname, as in JOBNAME(MAPES*).

When JOBNAME(jobname ...) is specified, by consecutive invocations of the SET command, the jobname list is deleted and rebuilt with the current specification each time the command is issued.

**ALLJOBNAMES**
- Specifies that tracing is to be in effect for all job names.

**NOJOBNAME**
- Specifies that use to resets tracing.

#### PDCALLABLE | NOPDCALLABLE

**PDCLASSABLE**
- Use to trace IBM Policy Director Authorization Services SAF calls.

**ALL | NONE | TYPE(type ...)**
- Use to control the degree of tracing.

**ALL**
- Use to enable tracing of all IBM Policy Director Authorization Services SAF calls.

**NONE**
- Use to reset tracing.

**TYPE(type ...)**
- Use to enable tracing of one or more specific IBM Policy Director Authorization Services SAF calls.
- The request types that are supported are listed in the following table:

<table>
<thead>
<tr>
<th>Callable service</th>
<th>Service / type number</th>
</tr>
</thead>
<tbody>
<tr>
<td>IRRSZA00</td>
<td>1</td>
</tr>
</tbody>
</table>
The TYPE operand is cumulative; issuing NOPDCALLABLE or PDCALLABLE(NONE) will reset the trace.

**NOPDCALLABLE**
Use NOPDCALLABLE to reset the trace; equivalent to PDCALLABLE(NONE).

**RACROUTE | NORACROUTE**

**RACROUTE**
Use to trace RACROUTE calls.

**ALL | NONE | TYPE(type ...)**
Use to control the degree of tracing.

- **ALL** Use to enable tracing of all RACROUTE calls.
- **NONE** Use to reset tracing.
- **TYPE(type ...)** Use to enable tracing of one or more specific RACROUTE calls. The request types that are supported are listed in the following table:

<table>
<thead>
<tr>
<th>RACROUTE REQUEST type</th>
<th>Service / type number</th>
</tr>
</thead>
<tbody>
<tr>
<td>AUTH</td>
<td>1</td>
</tr>
<tr>
<td>FASTAUTH</td>
<td>2</td>
</tr>
<tr>
<td>LIST</td>
<td>3</td>
</tr>
<tr>
<td>DEFINE</td>
<td>4</td>
</tr>
<tr>
<td>VERIFY</td>
<td>5</td>
</tr>
<tr>
<td>EXTRACT</td>
<td>6</td>
</tr>
<tr>
<td>DIRAUTH</td>
<td>7</td>
</tr>
<tr>
<td>TOKENMAP</td>
<td>8</td>
</tr>
<tr>
<td>VERIFYX</td>
<td>9</td>
</tr>
<tr>
<td>TOKENXTR</td>
<td>10</td>
</tr>
<tr>
<td>TOKENBLD</td>
<td>11</td>
</tr>
<tr>
<td>EXTRACT, BR=YES</td>
<td>12</td>
</tr>
<tr>
<td>AUDIT</td>
<td>13</td>
</tr>
<tr>
<td>STAT</td>
<td>14</td>
</tr>
<tr>
<td>SIGNON</td>
<td>15</td>
</tr>
<tr>
<td>TOKENMAP, XMEM</td>
<td>16</td>
</tr>
<tr>
<td>TOKENXTR, XMEM</td>
<td>17</td>
</tr>
</tbody>
</table>

The TYPE operand is cumulative; issuing NORACROUTE or RACROUTE(NONE) will reset the trace.

**NORACROUTE**
Use NORACROUTE to reset the trace; equivalent to RACROUTE(NONE).
SET

SYSTEMSSL | NOSYSTEMSSL

SYSTEMSSL
Use to trace RACF’s use of z/OS System Secure Sockets Layer (SSL) services. The actual trace records are created by the System SSL component itself; RACF only requests that the trace records be created. SSL services are used by RACF to create PKCS #7 envelopes for user passwords and password phrases. For more information, see z/OS Security Server RACF Security Administrator’s Guide.

When SET TRACE(SYSTEMSSL) is in effect, all trace functions except EBCDIC and ASCII data dumps are requested. The trace records are written to a UNIX file with the pathname /tmp/gskssl.racf.pid.trc, where pid is the UNIX process ID assigned to the RACF thread that attempted the enveloping operation (initial creation during password or password phrase change, or retrieval with the R_admin callable service). It is difficult to determine the pid for a given enveloping operation because it exists transiently. To debug a reproducible problem, look at the trace files that already exist in /tmp starting with gskssl.racf, or delete all the trace files, initiate the enveloping operation, find the new file that was created, and look within that file for the trace data. For more information on creating trace records, see z/OS Cryptographic Services System SSL Programming.

NOSYSTEMSSL
Deactivates tracing for RACF’s use of System SSL services.

Examples

Table 93. SET examples

<table>
<thead>
<tr>
<th>Example 1</th>
<th>Operation</th>
<th>User ADMIN wants to enable automatic command direction and establish that LAURIE at POKMVS and the command issuer receives output and notification when an automatically directed command receives a return code of 8 or greater.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Known</td>
<td>The RACF subsystem prefix is @.</td>
</tr>
<tr>
<td></td>
<td>Command</td>
<td>@SET AUTODIRECT (OUTPUT (FAIL (POKMVS.LAURIE &amp;RACUID)) NOTIFY (FAIL (POKMVS.LAURIE &amp;RACUID)))</td>
</tr>
<tr>
<td></td>
<td>Defaults</td>
<td>None.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Example 2</th>
<th>Operation</th>
<th>User ADMIN wants to enable automatic command direction and establish that:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>• ACDEERROR at POKMVS will receive warning and error output, but no TSO SEND messages.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• ANDREW at POKMVS will receive warning output, error output, and TSO SEND messages for error conditions.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• LAURIE at POKMVS will not receive any output, but will receive TSO SEND messages for error conditions.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• The command issuer gets no notification of automatically directed commands.</td>
</tr>
<tr>
<td></td>
<td>Known</td>
<td>The RACF subsystem prefix is @. LAURIE at POKMVS has the ability to browse the RRSFLIST data set of ACDEERROR at POKMVS to determine what needs to be fixed.</td>
</tr>
<tr>
<td></td>
<td>Command</td>
<td>@SET AUTODIRECT (OUTPUT (WARN (POKMVS.ACDEERROR POKMVS.ANDREW)) NOTIFY (FAIL (POKMVS.LAURIE POKMVS.ANDREW)))</td>
</tr>
<tr>
<td></td>
<td>Defaults</td>
<td>None.</td>
</tr>
</tbody>
</table>
### Example 3

**Operation**
User ADMIN wants to enable automatic direction, automatic password direction, automatic direction of application updates, but not password synchronization. User ADMIN wants to be notified and receive output for all failures. The command issuer needs to always receive notification and output for automatically directed commands (but not for automatic password direction or automatic direction of application updates).

**Known**
The RACF subsystem prefix is `@`.

**Commands**
```bash
@SET AUTODIRECT(OUTPUT(FAIL(POKMVS.ADMIN &RACUID))
NOTIFY(FAIL(POKMVS.ADMIN &RACUID)))
@SET AUTOPWD(OUTPUT(FAIL(POKMVS.ADMIN))
NOTIFY(FAIL(POKMVS.ADMIN)))
@SET AUTOAPPL(OUTPUT(FAIL(POKMVS.ADMIN))
NOTIFY(FAIL(POKMVS.ADMIN)))
```

**Defaults**
None.

### Example 4

**Operation**
User ADMIN wants to enable tracing for all VERIFY requests issued in a particular address space.

**Known**
The RACF subsystem prefix is `@`.

**Command**
```bash
@SET TRACE(RACROUTE(TYPE(2,5,9)) ASID(17))
```

**Defaults**
None.

### Example 5

**Operation**
User ADMIN wants to obtain information concerning the RRSF node’s configuration and status related to the RACF subsystem. User ADMIN also wants to turn on tracing for IMAGE events.

**Known**
Since the LIST keyword is used in combination with the TRACE keyword, the information displayed reflects the results after processing the TRACE keyword. The RACF subsystem prefix is `@`.

**Command**
```bash
@SET LIST TRACE(IMAGE)
```

**Defaults**
None.

### Example 6

**Operation**
User ADMIN wants to enable tracing for the aznAccess SAF callable service for job name IBMUSER.

**Known**
The RACF subsystem prefix `@`.

**Command**
```bash
@SET TRACE(PDCALLABLE(TYPE(1)) JOBNAME(IBMUSER))
```

**Defaults**
None.

### Example 7

**Operation**
User ADMIN wants to verify that tracing has been enabled for the aznAccess z/OS Policy Director SAF callable service.

**Known**
The RACF subsystem prefix is `@`.

**Command**
```bash
@SET LIST
```

**Defaults**
None.

**Output**
SET LIST output indicates that tracing has been enabled for PDCALLABLE service request type 1 (aznAccess SAF callable service). See Figure 73.
RACFR12 IRRH005I (R) RACF SUBSYSTEM INFORMATION:
  TRACE OPTIONS - IMAGE
    - NOAPP
    - SYSTEMSSL
    - RACROUTE
      2 5 9
    - NOCALLABLE
    - PDCLASS
      1
    - NODATABASE
    - ASID
      17
    - JOBNAME
      IBMUSER
  SUBSYSTEM USERID - IBMUSER
  JESNODE (FOR TRANSMITS) - POKVMMCL
  AUTOMATIC COMMAND DIRECTION IS *NOT* ALLOWED
  AUTOMATIC PASSWORD DIRECTION IS *NOT* ALLOWED
  PASSWORD SYNCHRONIZATION IS *NOT* ALLOWED
  AUTOMATIC DIRECTION OF APPLICATION UPDATES IS *NOT* ALLOWED
  RACF STATUS INFORMATION:
    TEMPLATE VERSION - HRF7709 00000020.00000030
    DYNAMIC PARSE VERSION - HRF7709

Figure 73. Output for SET LIST Command
SETROPTS

SETROPTS (Set RACF options)

Purpose

Use the SETROPTS command to set system-wide RACF options related to resource protection dynamically. Specifically, you can use SETROPTS to do the following:

- Gather and display RACF statistics
- Protect terminals
- Log RACF events
- Permit list-of-groups access checking
- Display options currently in effect
- Enable or disable the generic profile checking facility on a class-by-class basis
- Activate checking for previous passwords and password phrases
- Limit unsuccessful attempts to access the system using incorrect passwords and password phrases
- Control change intervals for passwords and password phrases
- Control mixed-case passwords
- Warn of expiring passwords and password phrases
- Establish password syntax rules
- Activate auditing for access attempts by class
- Activate auditing for security labels
- Require that all work entering the system, including users logging on and batch jobs, have a security label assigned
- Enable or disable the global access checking facility
- Refresh in-storage profile lists and global access checking tables
- Set the password the operator must supply in order for RACF to complete an RVARY command that changes RACF status or changes the RACF databases
- Enable or disable the sharing, in common storage, of discrete and generic profiles for general resource classes
- Activate or deactivate auditing of access attempts to RACF-protected resources based on installation-defined security levels
- Control the automatic data set protection (ADSP) attribute for users
- Activate profile modeling for GDG, group, and user data sets
- Activate protection for data sets with single-level names
- Control logging of real data set names
- Control the job entry subsystem options
- Activate tape data set protection
- Control whether RACF is to allow users to create or access data sets that do not have RACF protection
- Activate and control the scope of erase-on-scratch processing
- Activate program control, which includes both access control to load modules and program access to data
- Prevent users from accessing uncataloged permanent data sets
- Establish a system-wide VTAM session interval
- Set an installation-wide default for the RACF security retention period for tape data sets
• Activate enhanced generic naming for data sets and entries in the global access checking table
• Set installation defaults for primary and secondary national languages
• Activate auditing for APPC transactions
• Use the dynamic class descriptor table.

If you specify the AUDIT operand, RACF logs all uses of the RACROUTE REQUEST=DEFINE SVC and all changes made to profiles by RACF commands.

Following are the classes that can be specified in the AUDIT operand and the commands and SVCs that are logged for each class.

<table>
<thead>
<tr>
<th>USER</th>
<th>GROUP</th>
<th>DATASET</th>
<th>CDT entries</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADDUSER</td>
<td>ADDGROUP</td>
<td>ADDSD</td>
<td>PERMIT</td>
</tr>
<tr>
<td>ALTUSER</td>
<td>ALTGROUP</td>
<td>ALTDSO</td>
<td>REQUEST=DEFINE SVC</td>
</tr>
<tr>
<td>CONNECT</td>
<td>CONNECT</td>
<td>DELDSO</td>
<td>RALTER</td>
</tr>
<tr>
<td>DELUSER</td>
<td>DELGROUP</td>
<td>PERMIT</td>
<td>RDEFINE</td>
</tr>
<tr>
<td>PASSWORD</td>
<td>REMOVE</td>
<td>REQUEST=DEFINE SVC</td>
<td>RDELETE</td>
</tr>
<tr>
<td>REMOVE</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
</tbody>
</table>

Most RACF functions do not require special versions or releases of the operating system or operating system components. However, some do require that your system be at a certain level.

Using SETROPTS when RACF is enabled for sysplex communication: When RACF is enabled for sysplex communication, RACF propagates the following SETROPTS commands:
• GENERIC REFRESH
• GLOBAL
• GLOBAL REFRESH
• RACLST
• NORACLST
• RACLST REFRESH
• WHEN(PROGRAM)
• WHEN(PROGRAM) REFRESH

When issued from a member of the RACF data sharing group, these commands, if successful on the member that issues them, are propagated in a controlled, synchronized manner to the other members in the group. A system in read-only mode can participate if it receives a SETROPTS command propagated from another system, but a user on a system in read-only mode cannot issue any SETROPTS commands except for the SETROPTS LIST command. For propagated SETROPTS REFRESH commands, members of the data sharing group are notified to either create, update, or delete some in-storage information. These commands are coordinated to ensure that all systems begin to use the changed information simultaneously, and to always see a consistent view of this information.

RACF serializes propagated SETROPTS commands to prevent conflicting commands of the same type (for example, SETROPTS RACLST and SETROPTS NORACLST) from processing simultaneously.
SETROPTS

Refer to the specific parameter descriptions for additional information about using these parameters.

Notes:

1. The options you specify on SETROPTS are common on systems that share the RACF database. All the systems involved must have the required levels of software. If you activate SECLABEL and the multilevel security options on one system, they are activated on all systems.

2. If RACF is not enabled for sysplex communication, the SETROPTS commands that would be propagated to all members of a data sharing group must instead be issued on each system sharing the database. Although the command is not propagated, RACF does record the fact that a SETROPTS RACLST was issued. The next time that any system sharing the database is IPLed, the SETROPTS RACLST is done on that sharing system.

3. When the SETROPTS command is from ISPF, the TSO command buffer (including password data) is written to the ISPLOG data set. As a result, you should not issue the SETROPTS command from ISPF or you must control the ISPLOG data set carefully.

4. If the SETROPTS command is issued as a RACF operator command, the command and the password data is written to the system log. Therefore, use of SETROPTS as a RACF operator command should either be controlled or you should issue the command as a TSO command.

RACF date handling: RACF interprets dates with 2-digit years as follows. (The yy value represents the 2-digit year.)

- If \( 70 < yy \leq 99 \), the date is interpreted as 19yy.
- If \( 00 \leq yy \leq 70 \), the date is interpreted as 20yy.

Issuing options

The following table identifies the eligible options for issuing the SETROPTS command:

<table>
<thead>
<tr>
<th>As a RACF TSO command?</th>
<th>As a RACF operator command?</th>
<th>With command direction?</th>
<th>With automatic command direction?</th>
<th>From the RACF parameter library?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes (See rule.)</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Rule: The SETROPTS LIST command without other keywords is not eligible for automatic command direction.

For information on issuing this command as a RACF TSO command, refer to Chapter 3, “RACF TSO commands,” on page 15.

For information on issuing this command as a RACF operator command, refer to Chapter 4, “RACF operator commands,” on page 21.

You must be logged on to the console to issue this command as a RACF operator command.

Authorization required

When issuing the SETROPTS command as a RACF operator command, you might require sufficient authority to the proper resource in the OPERCMDS class. See z/OS Security Server RACF Security Administrator’s Guide for further information.
Most SETROPTS command functions require you to have the SPECIAL or AUDITOR attributes.

If you have the SPECIAL attribute you can use all of the operands except those listed below that require the AUDITOR attribute:

- APPLAUDIT | NOAPPLAUDIT
- AUDIT | NOAUDIT
- CMDVIOL | NOCMDVIOL
- LOGOPTIONS
- OPERAUDIT | NOOPERAUDIT
- SAUDIT | NOSAUDIT
- SECLABELAUDIT | NOSECLABELAUDIT
- SECLEVELAUDIT | NOSECLEVELAUDIT

If you have either the SPECIAL or AUDITOR attribute, you can use the LIST operand.

To specify the AT keyword, you must have READ authority to the DIRECT.node resource in the RRSFDATA class and a user ID association must be established between the specified node.userid pair(s).

To specify the ONLYAT keyword you must have the SPECIAL attribute, the userid specified on the ONLYAT keyword must have the SPECIAL attribute, and a user ID association must be established between the specified node.userid pair(s) if the user IDs are not identical.

In some situations, you can use SETROPTS even if you do not have the SPECIAL or AUDITOR attributes. These situations are:

- You can specify the LIST operand if you have the group-SPECIAL or group-AUDITOR attribute in the current connect group or if GRPLIST is active in any group that you are connected to.
- You can specify REFRESH together with GENERIC if you have the group-SPECIAL, AUDITOR, group-AUDITOR, OPERATIONS, group-OPERATIONS attribute, or CLAUTH authority for the classes specified.
- You can specify REFRESH together with GLOBAL if you have the OPERATIONS attribute or CLAUTH authority for the classes specified.
- You can specify REFRESH together with RACLST if you have CLAUTH authority to the specified class. You can specify REFRESH together with WHEN(PROGRAM) if you have CLAUTH authority for the program class.

Note: The syntax diagram does not indicate the defaults that are in effect when RACF is using a newly initialized database. You can find these defaults in the description of each operand. As you establish the system-wide defaults your installation needs, you might find it useful to mark the syntax diagram to reflect your choices.

Syntax

For the key to the symbols used in the command syntax diagrams, see Syntax of RACF commands and operands on page 9. The complete syntax of the SETROPTS command is:

[subsystem-prefix][SETROPTS | SETR]
Parameters

**subsystem-prefix**
Specifies that the RACF subsystem is the execution environment of the command. The subsystem prefix can be either the installation-defined prefix for RACF (1–8 characters) or, if no prefix has been defined, the RACF subsystem name followed by a blank. If the command prefix was registered with CPF, you can use the MVS command D OPDATA to display it or you can contact your RACF security administrator.

Only specify the subsystem prefix when issuing this command as a RACF operator command. The subsystem prefix is required when issuing RACF operator commands.

**ADDCREATOR | NOADDCREATOR**

**ADDCREATOR**
Specifies that if a user defines any new DATASET or general resource profile using ADDSD, RDEFINE or RACROUTE REQUEST=DEFINE, the profile creator’s user ID is placed on the profile access list with ALTER authority.
NOADDCREATOR
Specifies that if a user defines any new DATASET or general resource profile using ADDSD, RDEFINE or RACROUTE REQUEST=DEFINE, or creates discrete profiles other than DATASET and TAPEVOL using RACROUTE REQUEST=DEFINE, RACF does not place the profile creator's user ID on the profile's access list. If the profile creator uses profile modeling, RACF copies the access list exactly. If the creator's user ID appears in the model's access list, RACF copies the authority to the new profile. For example, if the creator's user ID appears in the model's access list with READ, RACF copies that access authority to the new profile without changing it to ALTER.

An important exception for NOADDCREATOR occurs when the user creates a discrete DATASET or TAPEVOL profile using RACROUTE REQUEST=DEFINE. In this case, RACF ignores the NOADDCREATOR options and places the profile creator's user ID on the new profile's access list with ALTER authority. If the profile creator uses profile modeling to define a discrete DATASET or TAPEVOL and the creator's user ID appears in the model's access list, RACF creates the authority in the new profile with ALTER authority. This exception to NOADDCREATOR allows system components to allocate data sets and immediately access them without having an administrator manipulate the profile's access list in the interim.

Note: The initial setting of the ADDCREATOR/NOADDCREATOR keyword depends on whether your database is new or old. When IRRMIN00 is run with PARM=NEW, the initial setting is NOADDCREATOR. When IRRMIN00 is run with anything other than PARM=NEW, RACF retains the current value of ADDCREATOR/NOADDCREATOR. For compatibility and migration reasons, this value is set to ADDCREATOR if no prior specification of ADDCREATOR or NOADDCREATOR had occurred.

ADSP | NOADSP
ADSP
Specifies that data sets created by users who have the automatic data set protection (ADSP) attribute is RACF-protected automatically.
ADSP is in effect when RACF is using a newly initialized database.
Because ADSP forces the creation of a discrete profile for each data set created by users who have the ADSP attribute, you should normally specify NOADSP if you specify GENERIC.

NOADSP
Cancels automatic RACF protection for users who have the ADSP attribute.
Because ADSP forces the creation of a discrete profile for each data set created by users who have the ADSP attribute, you should normally specify NOADSP if you specify GENERIC.

APPLAUDIT | NOAPPLAUDIT
APPLAUDIT
Specifies that auditing of APPC transactions on your system be enabled. APPC transactions are audited when they receive authorization (start) or have authorization removed (end). You must request auditing for the appropriate APPL profile. Otherwise, turning APPLAUDIT on does not cause auditing of APPC transactions. See z/OS Security Server RACF Auditor's Guide for more information on requesting auditing.
SETROPTS

You must have the AUDITOR attribute to specify this option.

NOAPPLAUDIT

Specifies that auditing of APPC transactions on your system (starting and ending) be disabled. You must have the AUDITOR attribute to specify this option.

AT | ONLYAT

The AT and ONLYAT keywords are only valid when the command is issued as a RACF TSO command.

AT([node].userid ...)

Specifies that the command is to be directed to the node specified by node, where it runs under the authority of the user specified by userid in the RACF subsystem address space.

If node is not specified, the command is directed to the local node.

ONLYAT([node].userid ...)

Specifies that the command is to be directed only to the node specified by node where it runs under the authority of the user specified by userid in the RACF subsystem address space.

If node is not specified, the command is directed only to the local node.

Note: SETROPTS LIST with no other keywords specified is not eligible for automatic command direction. Do not specify the ONLYAT and LIST keywords together without any other keywords on a SETROPTS command.

AUDIT | NOAUDIT

AUDIT(class-name ... I *)

Specifies the names of the classes for which you want RACF to perform auditing. For the classes you specify, RACF logs all uses of the RACROUTE REQUEST=DEFINE SVC and all changes made to profiles by RACF commands. When the class specified is USER, RACF logs all password and password phrase changes made by RACROUTE REQUEST=VERIFY. (RACF adds the classes you specify to those already specified for auditing.)

The valid class names are USER, GROUP, DATASET, and those defined in the class descriptor table. For a list of general resource classes defined in the class descriptor table supplied by IBM, see Appendix B, “Supplied RACF resource classes,” on page 659.

If you specify an asterisk (*), logging occurs for all classes.

You must have the AUDITOR attribute to enter the AUDIT operand.

Note: If you activate auditing for a class using SETROPTS AUDIT, RACF activates auditing for all classes in the class descriptor table that have the same POSIT value as the class you specify. For example, the classes TIMS, GIMS, and AIMS all have a POSIT value of 4 in their respective class descriptor table entries. If you activate auditing for any one of these classes, you activate auditing for all of them.

For more information on sharing a POSIT value, see the POSIT keyword of the RDEFINE command.

NOAUDIT(class-name ... I *)

Specifies the names of the classes for which you no longer want RACF to
SETROPTS

perform auditing. For the classes you specify, RACF no longer logs all uses of the REQUEST=DEFINE SVC and all changes made to profiles by RACF commands. The valid class names are USER, GROUP, DATASET, and those classes defined in the class descriptor table. For a list of general resource classes defined in the class descriptor table supplied by IBM, see Appendix B, “Supplied RACF resource classes,” on page 659.

NOAUDIT(\*) is in effect when RACF is using a newly initialized database. If you specify an asterisk (\*), logging occurs for any of the classes.

You must have the AUDITOR attribute to enter the NOAUDIT operand.

**Note:** If you deactivate auditing for a class using SETROPTS NOAUDIT, RACF deactivates auditing for all classes in the class descriptor table that have the same POSIT value as the class you specify. For example, the classes TIMS, GIMS, and AIMS all have a POSIT value of 4 in their respective class descriptor table entries. If you deactivate auditing for any one of these classes, you deactivate auditing for all of them.

For more information on sharing a POSIT value, see the POSIT keyword of the RDEFINE command.

CATDSNS | NOCATDSNS

**CATDSNS (FAILURES | WARNING)**

Specifies that uncataloged data sets, new (and not cataloged), or system temporary data sets are not to be accessed by users.

The following exceptions apply:

1. The job that creates the data set can access it even if the data set is uncataloged. If the data set is still uncataloged when the job ends, it is inaccessible thereafter.
2. Data sets with discrete profiles can be accessed—even if uncataloged—if allowed by the profile.
3. For uncataloged data sets without discrete profiles, RACF constructs a resource name of ICHUNCAT.dsnname (only the first 30 characters of the dsnname is used). It checks the user’s authority to this resource in the FACILITY class. If the resource is protected by a FACILITY class profile, and the user has access to it, the access is allowed.
4. If the user has the SPECIAL attribute, the access is allowed even if the data set is uncataloged, but a warning message and SMF record is created.
5. If you use DFSMSrmm™ to manage your tape data sets and the TAPEAUTHF1 option is active (in the DEVSUPxx member of SYS1.PARMLIB), an uncataloged tape data set might be read by a user who has access to the first file on the tape volume when the first file is cataloged. See [z/OS DFSMSrmm Implementation and Customization Guide](https://www.ibm.com/support/knowledgecenter/SS1EDY_1.12.1/guide/rmm_sdfmstg.html) (If you use a different tape management system, refer to your product documentation.
6. Write requests to tape data sets are not denied because of SETROPTS CATDSNS.

CATDSNS might have a negative impact on RACF and system performance because RACF must verify that data sets are cataloged before it allows them to be opened.
Note: For additional information about accessing uncataloged data sets, refer to SETROPTS command in z/OS Security Server RACF Security Administrator’s Guide.

FAILURES Specifies that RACF is to reject any request to access a data set that is not cataloged.

FAILURES is the default.

If CATDSNS(FAILURES) is in effect and a privileged started task or a user with the SPECIAL attribute requests access of an uncataloged data set, RACF accepts the request and issues a warning message.

WARNING Specifies that the access is allowed even if the data set is uncataloged. However, a warning message and SMF record is created.

NOCATDSNS Specifies that data sets that are not cataloged can be accessed by users.

NOCATDSNS is in effect when RACF is using a newly initialized database.

CLASSACT | NOCLASSACT

CLASSACT(class-name ... | *)
Specifies those classes defined by entries in the class descriptor table for which RACF protection is to be in effect.

If you specify an asterisk (*), you activate RACF protection for all classes defined in the class descriptor table except for those classes with a default return code of 8. For a list of general resource classes defined in the class descriptor table supplied by IBM, see Appendix B, “Supplied RACF resource classes,” on page 659.

Notes:
1. If you activate a class using SETROPTS CLASSACT, RACF activates all classes in the class descriptor table that have the same POSIT value as the class you specify. For example, the classes TIMS, GIMS, and AIMS all have a POSIT value of 4 in their respective class descriptor table entries. If you activate any one of these classes, you activate all of them.

   For more information on sharing a POSIT value, see the POSIT keyword of the RDEFINE command.

2. Before activating a class that has a default return code of 8 in the class descriptor table (either explicitly or by means of a shared POSIT value), be sure you have defined the necessary profiles to allow your users to access resources in that class. For example, if you activate JESINPUT without defining profiles to allow access, no one is able to submit batch jobs.

3. You need not activate the DIGTCERT, DIGTCRIT, and DIGTRING classes to use resources in those classes. However, performance is improved when you RACLIST the DIGTCERT and DIGTCRIT classes if you use resources in these classes. To RACLIST a class, you must activate it.

NOCLASSACT(class-name ... | *)
Specifies those classes defined by entries in the class descriptor table for which RACF protection is not to be in effect. If you specify an asterisk (*), you deactivate RACF protection for all classes defined in the class
SETROPTS

descriptor table. For a list of general resource classes defined in the class descriptor table supplied by IBM, see Appendix B, “Supplied RACF resource classes,” on page 659.

NOCLASSACT is in effect when RACF is using a newly initialized database.

Rules:

• If you deactivate a class using SETROPTS NOCLASSACT, RACF deactivates all classes in the class descriptor table that have the same POSIT value as the class you specify. For example, the classes TIMS, GIMS, and AIMS all have a POSIT value of 4 in their respective class descriptor table entries. If you deactivate any one of these classes, you deactivate all of them.

For more information on sharing a POSIT value, see the POSIT keyword of the RDEFINE command.

• If MLACTIVE, MLS, MLIPCOBJ, MLFSOBJ or SECLBYSYSTEM is active, you may not deactivate the SECLABEL class. Issuing SETROPTS NOCLASSACT(SECLABEL) will fail.

CMDVIOL | NOCMDVIOL

Specifies whether RACF is to log violations detected by RACF commands. You must have the AUDITOR attribute to specify these options.

CMDVIOL

Specifies that RACF is to log violations detected by RACF commands (except LISTDSD, LISTGRP, LISTUSER, RLIST, and SEARCH) during RACF command processing. A violation might occur because a user is not authorized to modify a particular profile or is not authorized to enter a particular operand on a command.

CMDVIOL is in effect when RACF is using a newly initialized database.

NOCMDVIOL

Specifies that RACF is not to log violations detected by RACF commands during RACF command processing (except RVARY and SETROPTS, which are always logged).

COMPATMODE | NOCOMPATMODE

COMPATMODE

Allows users and jobs not using security labels to be on a system enforcing security labels. The ACEEs of the user IDs or jobs must have been created by a RACROUTE REQUEST=VERIFY that did not specify the RELEASE=1.9 keyword (or later).

NOCOMPATMODE

Users and jobs must be running with correct security labels to access data.

NOCOMPATMODE is in effect when RACF is using a newly initialized database.

EGN | NOEGN

Specifies whether or not to activate or deactivate enhanced generic naming (EGN).

EGN

Activates EGN. When you activate this option, RACF allows you to specify the generic character ** (in addition to the generic characters * and %) when you define data set profile names and entries in the global access checking table.
Notes:
1. EGN changes the meaning of the generic character *.
2. When you first activate enhanced generic naming, the RACF-protection provided by existing data set profiles and global access checking table remains the same.

For information on EGN and its effect on profile names, see the description of generic profiles in Appendix A, “Naming considerations for resource profiles” z/OS Security Server RACF Command Language Reference.

NOEGN
Specifies deactivation of EGN. When you deactivate this option, RACF does not allow you to specify the generic character ** when you define data set names and entries in the global access checking table.

NOEGN is in effect when RACF is using a newly initialized database.

---

Important
If you protect data sets with generic profiles while EGN is active and then deactivate this option, your resources can no longer be protected.

Table 101 on page 651 and Table 102 on page 651 show examples of generic profiles created with enhanced generic naming active.

Some of these profiles do not provide RACF protection when the option is deactivated. If a data set is unprotected when EGN is deactivated, you can protect the data set with a discrete profile—as described in Appendix A, “Naming considerations for resource profiles” z/OS Security Server RACF Command Language Reference—either before or after the option is deactivated, or with a generic profile after the option is deactivated.

---

ERASE | NOERASE

ERASE(erase-indicator)
Specifies that data management is to physically erase the contents of deleted data sets and scratched or released DASD extents. Erasing the data set means overwriting its contents with binary zeroes so that it cannot be read.

Restriction: The ERASE option applies to DASD data sets only, not tape data sets, unless you set the TAPEAUTHDSN option in the DEVSUPxx member of SYS1.PARMLIB. See “Erasing Scratched or Release Data (ERASE Option)” in z/OS Security Server RACF Security Administrator’s Guide for more information. For details about customizing SYS1.PARMLIB, see z/OS MVS Initialization and Tuning Reference. For details about controlling authorization for tape volume overwriting, see z/OS DFSMSdsm Implementation and Customization Guide. (If you use a different tape management system, refer to your product documentation.)

If you specify ERASE without any suboperand, whether a scratched data set is erased depends on the status of the erase indicator in the data set profile. The SETROPTS ERASE suboperand allow you to override the erase indicator in the data set profile, to control the scope of erase-on-scratch on an installation level rather than leaving it to individual users.

The SETROPTS ERASE erase-indicator can be:
ALL
Specifications that data management is to erase all scratched data sets, including temporary data sets, regardless of the erase indicator, if any, in the data set profile.

SECLEVEL\(\text{seclevel-name}\)
Specifications that data management is to erase all scratched data sets that have a security level equal to or greater than the security level that you specify, where \(\text{seclevel-name}\) must be a member of the SECLEVEL profile in the SECDATA class.

Note: A scratched data set with a security level lower than the level you specify is not erased unless the erase indicator (if any) in the data set profile is on.

NOSECLEVEL
Specifies that RACF is not to consider the security level in the data set profile when it decides whether data management is to erase a scratched data set.

Note: A scratched data set, regardless of security level, is not erased unless the erase indicator (if any) in the data set profile is on.

NOSECLEVEL is the default if you do not specify \(\text{erase-indicator}\) when you specify ERASE.

NOERASE
Specifications that erase-on-scratch processing is not in effect. NOERASE means that no data sets are erased when deleted (scratched), even if the erase indicator in the data set profile is on.

NOERASE is in effect when RACF is using a newly initialized database.

GENCMD | NOGENCMD

GENCMD\(\text{class-name} \ldots | \text{*}\)
Activates generic profile command processing for the specified classes. Valid class names are DATASET and all class names except grouping classes and classes defined with the GENERIC(DISALLOWED) attribute.

If you specify an asterisk (*), you activate generic profile command processing for the DATASET class plus all general resource classes except grouping classes and classes defined with the GENERIC(DISALLOWED) attribute.

Tip: For a list of classes defined on your system with GENERIC(DISALLOWED), issue the RLST CDT * CDTINFO command to list all classes defined in the dynamic CDT.

For a list of general resource classes supplied in the static CDT, see Appendix B, “Supplied RACF resource classes,” on page 659

When GENCMD is in effect for a class, all the command processors can work on generic profiles, but the RACF SVC routines cannot perform generic profile checking. This operand allows the installation to temporarily disable generic profile checking (during maintenance, for example) and still use the RACF commands to maintain generic profiles.

Generic profile command processing is automatically activated for all classes for which generic profile checking is activated. Therefore, when you issue SETROPTS GENERIC for a class, you need not issue SETROPTS GENCMD for the same class.
SETROPTS

**Note:** If you activate generic profile command processing for a class using SETROPTS GENCMD, RACF activates generic profile command processing for all classes in the class descriptor table that have the same POSIT value as the class you specify, except grouping classes. For example, the resource classes TIMS and AIMS and the grouping class GIMS all have a POSIT value of 4 in their respective class descriptor table entries. If you activate generic profile command processing for TIMS, you also activate it for AIMS. However, you cannot activate this option for GIMS because GIMS is a grouping class.

For more information on sharing a POSIT value, see the POSIT keyword of the RDEFINE command.

**NOGENCMD**(class-name ... | *)

Deactivates generic profile command processing for the specified classes. Valid class names are DATASET and all class names except grouping classes and classes defined with the GENERIC(DISALLOWED) attribute.

If you specify an asterisk (*), you deactivate generic profile command processing for the DATASET class plus all general resource classes except grouping classes and classes defined with the GENERIC(DISALLOWED) attribute.

**Tip:** For a list of classes defined on your system with GENERIC(DISALLOWED), issue the RLIST CDT * CDTINFO command to list all classes defined in the dynamic CDT.

For a list of general resource classes supplied in the static CDT, see Appendix B, “Supplied RACF resource classes,” on page 659.

NOGENCMD(*) is in effect when RACF is using a newly initialized database.

If generic profile checking is active (GENERIC is in effect), RACF ignores this operand because GENERIC both includes and overrides generic profile command processing.

**Note:** If you deactivate generic profile command processing for a class using SETROPTS NOGENCMD, RACF deactivates generic profile command processing for all classes in the class descriptor table that have the same POSIT value as the class you specify, except grouping classes. For example, the resource classes TIMS and AIMS and the grouping class GIMS all have a POSIT value of 4 in their respective class descriptor table entries. If you deactivate generic profile command processing for TIMS, you also deactivate it for AIMS. However, GIMS is unaffected because it is a grouping class.

For more information on sharing a POSIT value, see the POSIT keyword of the RDEFINE command.

**GENERIC | NOGENERIC**

**GENERIC**(class-name ... | *)

Activates generic profile checking for the classes specified.

Valid class names are DATASET and all class names except grouping classes and classes defined with the GENERIC(DISALLOWED) attribute.
SETROPTS

**Note:** Avoid activating generic profile checking for the DIGTCERT or DIGTRING class.

For a list of classes defined on your system with GENERIC(DISALLOWED), issue the RLIST CDT * CDTINFO command to list all classes defined in the dynamic CDT.

For a list of general resource classes supplied in the static CDT, see Appendix B, "Supplied RACF resource classes," on page 659.

For a list of the attributes for each class supplied in the static CDT, including GENERIC=DISALLOWED, see z/OS Security Server RACF Macros and Interfaces.

**Guidelines:**

- When possible, use generic profiles to protect multiple resources and reduce administrative effort. Consider issuing SETROPTS GENERIC(classname) for the classes you use, so that generic profiles are usable in those classes.
- If you already have general resource profiles defined in your database, avoid issuing the SETROPTS GENERIC(+) command. This command activates generic profile checking for all classes except resource grouping classes and classes defined with the GENERIC(DISALLOWED) attribute. Some classes, such as DIGTCERT and DIGTRING, do not support generic profile checking. These and other classes might already have profile names that contain generic characters (\*, @, and %).
- If a class already has profile names that contain generic characters (\*, @, and %), avoid issuing the SETROPTS GENERIC(classname) command for that class. Enabling generic profile checking for such a class prevents RACF from using previously defined profiles that contain generic characters in the name.
- In general, once you activate generic profile checking for a class and define generic profiles, avoid deactivating it with the NOGENERIC operand. RACF will not use your previously defined generic profiles for authorization checking while NOGENERIC is in effect.

Generic profile command processing is automatically activated for all classes for which generic profile checking is activated. Therefore, when you issue SETROPTS GENERIC for a class, you need not issue SETROPTS GENCMD for the same class.

If you specify GENERIC with REFRESH, only those currently active and authorized classes are refreshed.

**Notes:**

1. If RACF is enabled for sysplex communication, RACF propagates SETROPTS GENERIC(class-name) REFRESH commands to other members of the data sharing group.
2. If RACF is not enabled for sysplex communication, a SETROPTS GENERIC(class-name) REFRESH command is effective only on the system where it is issued.
3. If you specify GENERIC, you should also specify NOADSP.
4. If you activate generic profile checking for a class using SETROPTS GENERIC, RACF activates generic profile checking for all classes in the class descriptor table that have the same POSIT value as the class you specify, except grouping classes. For example, the resource classes TIMS and AIMS and the grouping class GIMS all have a POSIT value of
4 in their respective class descriptor table entries. If you activate generic
profile checking for TIMS, you also activate it for AIMS. However, you
cannot activate this option for GIMS because GIMS is a grouping class.

For more information on sharing a POSIT value, see the POSIT
keyword of the RDEFINE command.

**NOGENERIC(class-name ... | *)**

Deactivates the generic profile checking facility for the classes specified.

**Guideline:** In general, once you activate generic profile checking for a class
and define generic profiles, avoid deactivating it with the NOGENERIC
operand. RACF will not use your defined generic profiles for authorization
checking while NOGENERIC is in effect.

Valid class names are DATASET and all class names except grouping
classes and classes defined with the GENERIC(DISALLOWED) attribute.

For a list of classes defined on your system with GENERIC(DISALLOWED),
issue the RLIST CDT * CDTINFO command to list all classes defined in the
dynamic CDT.

For a list of general resource classes supplied in the static CDT, see

For a list of the attributes for each class supplied in the static CDT,
including GENERIC=DISALLOWED, see z/OS Security Server RACF
Macros and Interfaces.

NOGENERIC (*) is in effect when RACF is using a newly initialized
database.

NOGENERIC does not automatically deactivate generic profile command
processing. Therefore, when you issue SETROPTS NOGENERIC for a
class, issue SETROPTS NOGENCMDB if you want to deactivate generic
profile command processing for the same class.

If you specify GENCMDB with NOGENERIC, users can issue RACF
commands to maintain generic profiles, but RACF does not use generic
profile checking during authorization checking.

If you specify NOGENCMDB with NOGENERIC, all generic profile command
processing is deactivated.

**Note:** If you deactivate generic profile checking for a class using
SETROPTS NOGENERIC, RACF deactivates generic profile
checking for all classes in the class descriptor table that have the
same POSIT value as the class you specify, except grouping
classes. For example, the resource classes TIMS and AIMS and the
grouping class GIMS all have a POSIT value of 4 in their respective
class descriptor table entries. If you deactivate generic profile
checking for TIMS, you also deactivate it for AIMS. However, GIMS
is unaffected because it is a grouping class.

For more information on sharing a POSIT value, see the POSIT
keyword of the RDEFINE command.

**GENERICOWNER | NOGENERICOWNER**

**GENERICOWNER**

Restricts creation of profiles in all general resource classes except the
PROGRAM class.
To create a profile that is more specific than any existing profile protecting
the same resource a user must:

- Have the SPECIAL attribute
- Be the owner of the existing profile
- Have the group-SPECIAL attribute if a group owns the profile
- Have the group-SPECIAL attribute if the owner of the profile is in the
  scope of the group.

Notes:
1. GENERICOWNER provides protection only when there is an existing
   (less-specific) profile protecting the resource.
2. A less-specific profile must end in *, ** or trailing % characters. A more
   specific profile is a profile that matches the less-specific profile name,
   character for character, up to the ending *, or **, or trailing % characters
   in the less-specific name. If the less-specific profile ends in %, the
   characters in the more specific profile that correspond to the contiguous
   trailing % characters must not be either * or . characters. For more
   information, see "Permitting profiles for GENERICOWNER classes" on
   page 656.
   For example: To allow USERX to RDEFINE A.B in the JESSPOOL
   class, you need profile A.* in the JESSPOOL class, which is owned by
   USERX. You also need profile **, owned by the system administrator, to
   prevent other CLAUTH users from being able to RDEFINE A.B.
3. GENERICOWNER does not prevent the creation of a more specific
   profile if the more specific profile is created in the grouping class and is
   specified on the ADDMEM operand. For example, profile A* exists in
   the TERMINAL class and is owned by a group for which user ELAINE
   does not have group-SPECIAL. If the GENERICOWNER option is in
   effect, user ELAINE cannot define a more specific profile in the member
   class (such as, RDEF TERMINAL AA*), but user ELAINE can define a
   profile if it is specified on the ADDMEM operand for the grouping class
   profile—such as RDEF GTERMINL profile-name ADDMEM(AA*).

NOGENERICOWNER
Cancels the restriction on the creation of profiles for general resources.
NOGENERICOWNER is in effect when RACF is using a newly initialized
database.

GENLIST | NOGENLIST

GENLIST(class-name ...)
Also see RACLST operand.
Activates the sharing of in-storage generic profiles for the classes specified.
When GENLIST is active for a class, the generic profiles for that class are
loaded into common storage (ECSA) instead of being resident in the private
storage (ELSQA) of each user who references the class. Before activating
GENLIST for a class, you should check with your system programmer to
determine if your system is configured with enough ECSA to contain the
profiles.

The z/OS Security Server RACF System Programmer's Guide contains
information about the amount of virtual storage required for generic profiles,
and other considerations about when to use RACLST or GENLIST.
SETROPTS

Generally, when you do not share the RACF database with RACF on a VM system, RACLIST provides the best performance with the lowest usage of common storage.

The following classes supplied by IBM can be used with GENLIST:

<table>
<thead>
<tr>
<th>Class</th>
<th>Class</th>
<th>Class</th>
<th>Class</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>APPL</td>
<td>FACILITY</td>
<td>LOGSTRM</td>
<td>VMBATCH</td>
<td>VMRDR</td>
</tr>
<tr>
<td>CPSMOBJ</td>
<td>FIELD</td>
<td>RRSFDATA</td>
<td>VMCMD</td>
<td>VMSEGMT</td>
</tr>
<tr>
<td>DASDVOL</td>
<td>INFOMAN</td>
<td>SDSF</td>
<td>VNLAN</td>
<td></td>
</tr>
<tr>
<td>DCEUIDS</td>
<td>JESJOBS</td>
<td>TERMINAL</td>
<td>VMMDISK</td>
<td></td>
</tr>
<tr>
<td>DSNR</td>
<td>KEYSMSTR</td>
<td>TMEADM</td>
<td>VMNODE</td>
<td></td>
</tr>
</tbody>
</table>

When you activate GENLIST processing for a class, a generic profile in that class is copied from the RACF database into common storage the first time an authorized user requests access to a resource protected by the profile. The profile is retained in common storage and is available for all authorized users, thus saving real storage because the need to retain multiple copies of the same profile (one copy for each requesting user) in common storage is eliminated. Also, because RACF does not have to retrieve the profile each time a user requests access to a resource protected by it, this function saves processing overhead.

If you want to refresh shared in-storage generic profiles for a specific resource class, issue the SETROPTS command with the GENERIC(class-name) and REFRESH operands.

**Note:** RACF does not allow you to specify SETROPTS GENLIST and SETROPTS RACLIST for the same general resource class.

For information on sharing a POSIT value, see the POSIT keyword of the RDEFINE command.

**NOGENLIST(class-name ...)**

Also see NORACLIST operand.

Deactivates the sharing of in-storage generic profiles for the classes specified. Deactivate this function for general resource classes defined in the class descriptor table that are eligible for GENLIST processing. These classes are listed under the description for GENLIST.

When you specify NOGENLIST, RACF deletes in-storage generic profiles for the specified classes from common storage.

NOGENLIST is in effect for all classes defined in the class descriptor table when RACF is using a newly initialized database.

For information on sharing a POSIT value, see the POSIT keyword of the RDEFINE command.

**GLOBAL | NOGLOBAL**

**GLOBAL(class-name ... | *)**

Specifies those classes eligible for global access checking. If you specify an asterisk (*), you activate global access checking for all valid classes.

Valid classes you may specify are:

- The DATASET class
- The NODE grouping class
- The SECLABEL grouping class
SETROPTS

- All other classes defined in the class descriptor table, except for the remaining grouping classes.

For a list of general resource classes defined in the class descriptor table supplied by IBM, see Appendix B, “Supplied RACF resource classes,” on page 659.

If you specify GLOBAL with REFRESH, only those currently active and authorized classes are refreshed. If you have deleted the GLOBAL profile for a class, you should issue the SETROPTS command with the NOGLOBAL operand specified, rather than GLOBAL with REFRESH specified.

Notes:
1. If you activate global access checking for a class using SETROPTS GLOBAL, RACF activates global access checking for all classes in the class descriptor table that have the same POSIT value as the class you specify, except the excluded grouping classes. For example, the resource classes TIMS and AIMS and the grouping class GIMS all have a POSIT value of 4 in their respective class descriptor table entries. If you activate global access checking for TIMS, you also activate it for AIMS. However, you cannot activate this option for GIMS because GIMS is a grouping class.
   For more information on sharing a POSIT value, see the POSIT keyword of the RDEFINE command.
2. If RACF is enabled for sysplex communication, it propagates the SETROPTS GLOBAL and SETROPTS GLOBAL REFRESH commands to other systems in the sysplex if the command is successful on the system on which it was entered. If RACF is not enabled for sysplex communication, the command has to be issued on each system sharing the database.
3. Global access checking is bypassed if the user ID has the RESTRICTED attribute.

NOGLOBAL(class-name ... | *)
Deactivates global access checking for the specified classes. For more information on valid classes that are process by the NOGLOBAL operand, see the GLOBAL operand description.

NOGLOBAL(*) is in effect when RACF is using a newly initialized database.

Note: If you deactivate global access checking for a class using SETROPTS NOGLOBAL, RACF deactivates global access checking for all classes in the class descriptor table that have the same POSIT value as the class you specify, except for the excluded grouping classes. For example, the resource classes TIMS and AIMS and the grouping class GIMS all have a POSIT value of 4 in their respective class descriptor table entries. If you deactivate global access checking for TIMS, you also deactivate it for AIMS. However, GIMS is unaffected because it is a grouping class.
   For more information on sharing a POSIT value, see the POSIT keyword of the RDEFINE command.

GRPLIST | NOGRPLIST
GRPLIST
Specifies that authorization checking processing is to perform list-of-groups
access checking for all system users. When you specify GRPLIST, a user’s
authority to access or define a resource is not based only on the authority
of the user’s current connect group; access is based on the authority of any
group to which the user is connected.

**NOGRPLIST**
Specifies that the user’s authority to access a resource is based on the
authority of the user’s current connect group.

NOGRPLIST is in effect when RACF is using a newly initialized database.

**INACTIVE | NOINACTIVE**

**INACTIVE** *(unused-userid-interval)*
Specifies the number of days (1–255) that a user ID can remain unused
and still be considered valid. RACF user verification checks the number of
days since the last successful time the user accessed the system against
the INACTIVE value and, if the former is larger, revokes the user’s right to
use the system. INACTIVE applies to new users based on creation date. If
you specify INACTIVE, INITSTATS must be in effect.

If the backup database is needed but does not contain current information,
some user IDs can be revoked because they appear to have been unused
beyond the number of days specified on the INACTIVE operand. For more
information, see *z/OS Security Server RACF System Programmer’s Guide*.

**NOINACTIVE**
Specifies that RACF user verification is not to check user IDs against an
unused-userid-interval.

NOINACTIVE is in effect when RACF is using a newly initialized database.

**INITSTATS | NOINITSTATS**

**INITSTATS**
Specifies that statistics available during RACF user verification are to be
recorded. These statistics include the date and time the user was verified
by RACF, the number of user verifications that specified a particular group,
and the date and time of the user last requested verification with a
particular group. If you specify INACTIVE, REVOKE, HISTORY, or
WARNING, INITSTATS must be in effect.

For applications that specify the APPL operand on the RACROUTE
REQUEST=VERIFY macro, you can define a profile in the APPL class to
specify that the application needs only daily statistics recorded for its users.
To do this, specify the RACF-INITSTATS(DAILY) string in the APPLDATA field.
For more information about statistics collection, see *z/OS Security Server
RACF Security Administrator’s Guide*.

INITSTATS is in effect when RACF is using a newly initialized database.

**NOINITSTATS**
Specifies that statistics available during user verification are not to be
recorded.

**JES**
Controls job entry subsystem (JES) options. The JES options are:

**BATCHALLRACF | NOBATCHALLRACF**

**BATCHALLRACF**
Specifies that JES is to test for the presence of a user ID and password
SETROPTS

on the job statement or for propagated RACF identification information for all batch jobs. If the test fails, JES is to fail the job.

NOBATCHALLRACF
Specifies that JES is not to test for the presence of a user ID and a password on the statement, or propagated RACF identification information for all batch jobs.

NOBATCHALLRACF is in effect when RACF is using a newly initialized database.

EARLYVERIFY | NOEARLYVERIFY
This setting is ignored.

XBMALLRACF | NOXBMALLRACF

XBMALLRACF
Specifies that JES is to test for the presence of either a user ID and password on the JOB statement, or JES-propagated RACF identification information for all jobs to be run with an execution batch monitor. If the test fails, JES is to fail the job.

XBMALLRACF is only used on JES2.

NOXBMALLRACF
Specifies that JES is not to test for the presence of either a user ID and password on the JOB statement, or JES-propagated RACF identification information for all jobs to be run with an execution batch monitor.

NOXBMALLRACF is in effect when RACF is using a newly initialized database.

NJEUSERID(userid)
Defines the name (user ID) associated with SYSOUT or jobs that arrive through the network without an RTOKEN or UTOKEN.

The initial user ID (default user ID) after RACF data set initialization is ???????? (eight question marks).

Note: The variable userid cannot be a user ID defined in the RACF database. For more information, see the section on providing security for JES in z/OS Security Server RACF Security Administrator's Guide.

UNDEFINEDUSER(userid)
Defines the name (user ID) that is associated with local jobs that enter the system without a user ID.

The initial user ID (default user ID) after RACF data set initialization is ++++++++ (eight plus signs).

Note: The variable userid cannot be a user ID defined in the RACF database. For more information, see the section on providing security for JES in z/OS Security Server RACF Security Administrator's Guide.

KERBLVL
Specifies what level of key encryption processing should occur when a KERB segment is being processed for user and realm profiles. Beginning with z/OS Version 1 Release 9, the KERBLVL setting is ignored.
See z/OS Integrated Security Services Network Authentication Service Administration for information about how z/OS Network Authentication Service uses keys and how to customize environment variables related to keys.

**LANGUAGE**

Specifies the system-wide defaults for national languages (such as American English or Japanese) to be used on your system. You can specify a primary language, a secondary language, or both. The languages you specify depend on which products, when installed on your system, check for primary and secondary languages (using RACROUTE REQUEST=EXTRACT).

- If this user establishes an extended MCS console session, the languages you specify should be the same as the languages specified on the LANGUAGE LANGCODE statements in the MMSLSTxx PARMLIB member. See your MVS system programmer for this information.
- If this is a CICS user, see your CICS administrator for the languages supported by CICS on your system.

The SETROPTS LANGUAGE operand does not affect the language in which the RACF ISPF panels are displayed. The order in which the RACF ISPF panel libraries are allocated determines the language used. If your installation ordered a translated feature of RACF, the RACF program directory gives instructions for setting up the ISPF panels.

**PRIMARY**(*language*)

Specifies the installation’s default primary language.

The variable *language* can be a quoted or unquoted string.

If the PRIMARY suboperand is not specified, the primary language is not changed.

**SECONDARY**(*language*)

Specifies the installation’s default secondary language.

The language name can be a quoted or unquoted string.

If the SECONDARY suboperand is not specified, the secondary language is not changed.

**Notes:**

1. For both the PRIMARY and SECONDARY suboperands, specify the installation-defined name of a currently active language (a maximum of 24 characters) or one of the language codes (3 characters in length) that is installed on your system. For a list of valid codes, see *National Language Design Guide, Volume 2, National Language Support Reference Manual*, SE09-8002.
2. If the MVS message service is not active, the PRIMARY and SECONDARY values must be a 3-character language code.
3. The same language can be specified for both PRIMARY and SECONDARY.
4. RACF is shipped with both the primary and secondary language defaults set to ENU, meaning United States English.

**LIST**

Specifies that the current RACF options are to be displayed. If you specify operands in addition to LIST on the SETROPTS command, RACF processes the other operands before it displays the current set of options.

If RACF is enabled for sysplex communication and the system is in read-only mode, users on that system can issue the SETROPTS LIST command. All other operands are ignored.
SETROPTS

You must have the SPECIAL, AUDITOR, group-SPECIAL, or group-AUDITOR attribute to enter the LIST operand.

If you have the SPECIAL or group-SPECIAL attribute, RACF displays all operands except these auditing operands:

- APPLAUDIT | NOAPPLAUDIT
- AUDIT | NOAUDIT
- CMDVIOL | NOCMDVIOL
- LOGOPTIONS
- OPERAUDIT | NOOPERAUDIT
- SAUDIT | NOSAUDIT
- SECLABELAUDIT | NOSECLABELAUDIT.

If you have the AUDITOR or the group-AUDITOR attribute, RACF displays all operands.

**Note:** SETROPTS LIST with no other keywords specified is not eligible for automatic command direction. Do not specify the ONLYAT and LIST keywords together without any other keywords on a SETROPTS command.

**LOGOPTIONS** `(auditing-level class-name . . .) . . .`

Audits access attempts to resources in specified classes according to the auditing level specified. You must have the AUDITOR attribute. You can specify the DATASET class and any classes in the class descriptor table. The resources need not have profiles created in order for auditing to occur. See [z/OS Security Server RACF Auditor's Guide](#) for more information on when auditing occurs.

The SUCCESSES and FAILURES operands result in auditing in addition to any auditing specified in profiles in the class. In contrast, the ALWAYS and NEVER operands override any auditing specified in profiles in the class. Note that LOG=NONE, specified on a RACROUTE REQUEST=AUTH, takes precedence (auditing is not performed).

**auditing-level**

Specifies the access attempts to be logged for class-name. These options are processed in the order listed below. Thus, if class-name is specified with both SUCCESSES and ALWAYS in the same command, auditing takes place at the SUCCESSES level because option SUCCESSES is processed after ALWAYS.

**ALWAYS**
All access attempts to resources protected by the class are audited.

**NEVER**
No access attempts to resources protected by the class are audited. (All auditing is suppressed.)

**SUCCESSES**
All successful access attempts to resources protected by the class are audited.

**FAILURES**
All failed access attempts to resources protected by the class are audited.
**DEFAULT**

Auditing is controlled by the profile protecting the resource, if a profile exists. You can specify DEFAULT for all classes by specifying an asterisk (*) with DEFAULT.

LOGOPTIONS(DEFAULT) is in effect when RACF is using a newly initialized database.

**class-name**

The RACF class to which auditing-level applies. The class-name value can be DATASET and any classes in the class descriptor table. Each class can have only one auditing level associated with it. The auditing levels are processed in the following order:

1. ALWAYS
2. NEVER
3. SUCCESSES
4. FAILURES
5. DEFAULT

This processing order occurs independently of the order you specify the auditing levels. If you specify two or more auditing levels for a class in the same command, only the last option processed takes effect. Thus, if you specify the following command:

```
SETR LOGOPTIONS (FAILURES (DATASET, SECLABEL),
                 ALWAYS (DATASET, APPL),
                 DEFAULT (DATASET, GLOBAL))
```

The options in effect for the classes is:

- ALWAYS for the APPL class
- FAILURES for the SECLABEL class
- DEFAULT for the DATASET and GLOBAL classes

The DATASET and APPL classes are first assigned auditing-level ALWAYS. The DATASET class is then assigned auditing-level FAILURES, as is class SECLABEL. Finally, the DATASET class is assigned DEFAULT auditing-level, as is class GLOBAL.

If you specify one auditing-level for class-name and in a separate command specify a new auditing level for the same class name, the new auditing-level take effects.

SETROPTS LOGOPTIONS(DEFAULT(∗)) is in effect when RACF is using a newly initialized database.

For information on sharing a POSIT value, see the POSIT keyword of the RDEFINE command.

**MLACTIVE | NOMLACTIVE**


**MLACTIVE (FAILURES | WARNING)**

Causes security labels to be required on all work entering the system and on all resources defined to USER, DATASET, and all classes defined in the class descriptor table that require security labels.

Rules:
This option is available only if the SECLABEL class is active. Activation of MLACTIVE will fail if the SECLABEL class is not active or being activated by the command activating MLACTIVE.

With MLACTIVE, user tasks running in a server address space must have a security label that is equivalent to the address space’s security label.

**Data set and general resource profiles in WARNING mode:** A user or task can access a resource that is in WARNING mode and has no security label even when MLACTIVE(FAILURES) is in effect and the class requires security labels. The user or task receives a warning message and gains access. (A data set or general resource is in WARNING mode when you define or modify the profile that protects it and you specify the WARNING operand.)

**FAILURES**
Specifies that RACF is to reject any request to create or access any resource that requires a security label in the profile that protects it, and does not have one, and to reject any work entering the system that does not have a security label.

The only exception is if MLS(FAILURES) and MLACTIVE(FAILURES) are in effect, and a privileged started task or a user with the SPECIAL attribute and the SYSHIGH SECLABEL attempts to access a resource that requires a security label and does not have one. In this case, RACF allows the request as long as the request does not declassify data.

**WARNING**
Specifies that when a user requests access to a resource that does not have a security label and the resource belongs to a class that requires security labels, access is allowed but a warning is issued. Also, when work enters the system without a security label, access is allowed but a warning is issued.

MLACTIVE(WARNING) is the default value.

**NOMLACTIVE**
Allows work to enter the system without a security label and allows requests to access a resource that does not have a security label and the resource belongs to a class that requires security labels.

NOMLACTIVE is in effect when RACF is using a newly initialized database.

**MLFSOBJ**

**MLFSOBJ (ACTIVE | INACTIVE )**

**ACTIVE**
Specifies that security labels are required for files and directories. When the SECLABEL class is active, and MLFSOBJ is active, access to files and directories without security labels is denied except by trusted or privileged started tasks. This option cannot be activated if the SECLABEL class is not active.

If you do not specify ACTIVE or INACTIVE, MLFSOBJ(ACTIVE) is the default.

**INACTIVE**
Specifies that security labels are not required for files and directories.
SETROPTS

INACTIVE is in effect when RACF is using a newly initialized database.

MLPCOBJ

MLPCOBJ (ACTIVE | INACTIVE )

ACTIVE

Specifies that security labels are required for interprocess communication. When the SECLABEL class is active, and MLPCOBJ is active, access to semaphores, message queues and shared memory without associated security labels is denied except by trusted or privileged started tasks. This option cannot be activated if the SECLABEL class is not active.

If you do not specify ACTIVE or INACTIVE, MLPCOBJ(ACTIVE) is the default.

INACTIVE

Specifies that security labels are not required for interprocess communication.

INACTIVE is in effect when RACF is using a newly initialized database.

MLNAMES | NOMLNAMES

MLNAMES

Specifies that users are restricted to viewing only the names of files and directories that could be read from their current security label, and to viewing data set names that they have access to from their current security label. When MLNAMES is active, users listing catalogs or directories will not see names of resources that they cannot currently access.

NOMLNAMES

Specifies that users are not restricted to viewing only the names of files and directories that they cannot currently access.

If you do not specify MLNAMES or NOMLNAMES, NOMLNAMES is the default.

NOMLNAMES is in effect when RACF is using a newly initialized database.

MLQUIET | NOMLQUIET

For the relationships among SECLABEL, MLS, MLACTIVE, and MLQUIET, see z/OS Security Server RACF Security Administrator’s Guide.

MLQUIET

Allows only started tasks, console operators, or users with the SPECIAL attribute to log on, start new jobs, or access resources. Actions requiring user verification, resource access checking, or resource definition are available only to the security administrator (SPECIAL user), a trusted computer base job (as indicated in the token), or the console operator.

When this option is enabled, the system is in a tranquil state.

NOMLQUIET

Allows all users access to the system.

NOMLQUIET is in effect when RACF is using a newly initialized database.

MLS | NOMLS

For the relationships among SECLABEL, MLS, MLACTIVE, and MLQUIET, see z/OS Security Server RACF Security Administrator’s Guide.
SETROPTS

MLS (FAILURES | WARNING)
Prevents a user from declassifying data. In order to copy data, the security label of the target must encompass the security label of the source.

Rules:
• This option is available only if the SECLABEL class is active.
• Activation of MLS will fail if the SECLABEL class is not active or being activated by the command activating MLS.

FAILURES  Specifies that RACF is to reject any request to declassify data.

WARNING  Specifies that when a user attempts to declassify data, RACF is to allow the request but issue warning messages to the user and the security administrator.

MLS(WARNING) is the default value if you do not specify either FAILURES or WARNING.

NOMLS
Allows users to declassify data within the same CATEGORY.
NOMLS is in effect when RACF is using a newly initialized database.

MLSTABLE | NOMLSTABLE

MLSTABLE
Allows the installation to indicate that no one on the system is allowed to alter the security label of an object or alter the definition of the security label, unless MLQUIET is in effect.

NOMLSTABLE
Allows the alteration of security label definitions or the security labels within a profile without requiring MLQUIET to be in effect.
NOMLSTABLE is in effect when RACF is using a newly initialized database.

MODEL | NOMODEL

MODEL
Specifies, through the following suboperands, the model profile processing options. For information about automatic profile modeling, refer to the z/OS Security Server RACF Security Administrator's Guide.

GDG | NOGDG
Specifies that RACF should attempt to protect RACF-indicated members of a generation data group (GDG) using a base profile with the same name as the GDG data set base name. If a base profile exists for a particular RACF-indicated member, then RACF uses the base profile when determining whether the user can access or create the member. Otherwise, RACF uses, or creates, an individual profile for the model. MODEL(GDG) has no effect on GDG members that are protected by generic profiles.

NOGDG specifies that GDG members should not be treated specially by RACF; they are processed as any other data set would be.

GROUP | NOGROUP
Specifies that when creating a new profile for a group-named data set, RACF should check whether a model profile is specified in the group profile. If so, that model profile should be used to complete the definition of the new data set profile.
NOGROUP specifies that RACF should not use model profiles to complete the definition of new group-named data sets.

USER | NOUSER
Specifies that when creating a new profile for all user ID-named data sets, RACF should check whether a model profile is specified in the user profile. If so, that model profile should be used to complete the definition of the new data set profile.

NOUSER specifies that RACF should not use model profiles to complete the definition of new user ID-named data sets.

NOMODEL
Specifies that there is no model profile processing for GDG, GROUP, or USER data sets.

NOMODEL is in effect when RACF is using a newly initialized database.

OPERAUDIT | NOOPERAUDIT
Specifies whether RACF is to log all actions allowed only because a user has the OPERATIONS (or group-OPERATIONS) attribute. You must have the AUDITOR attribute to enter these operands.

OPERAUDIT
Specifies that RACF is to log all actions, such as accesses to resources and commands, allowed only because a user has the OPERATIONS or group-OPERATIONS attribute.

NOOPERAUDIT
Specifies that RACF is not to log the actions allowed only because a user has the OPERATIONS or group-OPERATIONS attribute.

NOOPERAUDIT is in effect when RACF is using a newly initialized database.

PASSWORD (suboperands)
Specifies options to monitor and check passwords and password phrases:

HISTORY | NOHISTORY

HISTORY(number-of-previous-values)
Specifies the number of previous passwords and password phrases (1–32) that RACF saves for each user and compares with each new intended value. For example, if you specify 12 for your HISTORY number, RACF saves up to 12 passwords and 12 password phrases for each user.

When RACF finds a match with a previous value, or with the current password or password phrase, RACF rejects the new intended value.

If you increase the HISTORY number, RACF saves and compares that number of passwords and password phrases to the new intended value. If you subsequently reduce the HISTORY number, any previous passwords and password phrases stored in the user profile in excess of the newly specified HISTORY number are not deleted and continue to be used for comparison.

For example, if you specify 12 for your HISTORY number and subsequently reduce it to 8, RACF compares the old passwords and password phrases 9–12 with the new intended value.

If you specify HISTORY, INITSTATS must be in effect.
SETROPTS

NOHISTORY
Specifies that new password and password phrase values are only compared with the current password or password phrase. If prior history information exists in the user profile, it is neither deleted nor changed.

NOHISTORY is in effect when RACF is using a newly initialized database.

INTERVAL(maximum-change-interval)
Specifies the maximum number of days (1–254) each user’s password and password phrase are valid. For example, if you specify 90 for your INTERVAL number, each user’s password is valid for 90 days and each user’s password phrase (if set) is valid for 90 days.

RACF uses the value you specify for maximum-change-interval as both:
- The default value for new users defined to RACF through the ADDUSER command.
- The upper limit for users who specify the INTERVAL operand on the PASSWORD command.

When a user logs on to the system, RACF compares this INTERVAL value (the system interval) with the interval value specified in the user’s profile (the user’s interval). RACF uses the lower of the two values to determine if the user’s password and password phrase have expired.

The initial default at RACF initialization is 30 days. The maximum change interval cannot be less than the minimum change interval set with the MINCHANGE keyword.

MINCHANGE(minimum-change-interval)
Specifies the number of days that must pass between a user’s password and password phrase changes. Acceptable values are 0–254 (days), providing the number of days between changes does not exceed the maximum change interval specified by the INTERVAL keyword. For example, if you specify 5 for your MINCHANGE number, users cannot change their passwords more than once in 5 days, nor can they change their password phrases (if assigned) more than once in 5 days.

The initial default is 0 days, allowing users to change their passwords and password phrases more than once on the same day.

Users can not change their own passwords and password phrases within the minimum change interval. However, you can use the ALTUSER command to change another user’s password within the minimum change interval if you have at least one of the following authorities:
- You have the SPECIAL attribute.
- The user is within the scope of a group in which you have the group-SPECIAL attribute.
- You are the owner of the user’s profile.
- You have at least CONTROL authority to the IRR.PASSWORD.RESET resource in the FACILITY class, and the other user does not have the SPECIAL, OPERATIONS, AUDITOR, or PROTECTED attribute.
- You have at least CONTROL access to an appropriate resource in the FACILITY class (IRR.PWRESET.OWNER.owner or IRR.PWRESET.TREE.owner), and both of the following conditions are also true:
  - The other user does not have the SPECIAL, OPERATIONS, AUDITOR, or PROTECTED attribute.
SETROPTS

- You are not excluded from altering the user by the
  IRR.PWRESET.EXCLUDE.excluded-user resource in the FACILITY
  class.

For more information about the IRR.PWSET profiles, see z/OS
Security Server RACF Security Administrator's Guide

MIXEDCASE | NOMIXEDCASE

MIXEDCASE
Indicates that all applications on this system and those that share the
RACF database support mixed-case and lowercase passwords. The
syntax rules must be modified to allow mixed-case and lowercase
characters. (See the RULE section on page 612 for more information.)
When this option is activated, the RACF ALTUSER, ADDUSER,
PASSWORD and RACLINK commands do not translate passwords to
uppercase, nor do applications that provide mixed-case password
support, such as TSO/E and z/OS UNIX Systems Services. This option
is inactive by default.

If you are propagating passwords with RRSF, see RRSF
Considerations for Mixed-Case Passwords in z/OS Security Server
RACF Security Administrator's Guide

Important: The MIXEDCASE option is intended to be activated—after
evaluating and updating applications and implementing appropriate
password syntax rules—and never deactivated. Deactivate it only if
problems are encountered. If you deactivate MIXEDCASE after it was
active, any users who changed their passwords to mixed or lower case
(when MIXEDCASE was active) will no longer be able to enter the
system until an authorized user resets their passwords to uppercase. If
you subsequently reactivate MIXEDCASE, the same users must enter
their passwords in upper case.

NOMIXEDCASE
Indicates that mixed-case and lowercase passwords are not supported.
This is the default setting.

Important: If you issue SETR NOMIXEDCASE after MIXEDCASE was
active, any users who changed their passwords to mixed-case or
lowercase (when MIXEDCASE was active) can no longer enter the
system until an authorized user resets their passwords to uppercase.
See the important note for the MIXEDCASE operand.

REVOKE | NOREVOKE

REVOKE(number-of-unsuccessful-attempts)
Specifies the number of consecutive unsuccessful attempts (1–255) to
access the system (using an incorrect password or password phrase)
before RACF revokes the user ID on the next unsuccessful attempt. If
you specify REVOKE, INITSTATS must be in effect.

The REVOKE number you specify applies to the combination of
incorrect passwords and password phrases RACF allows. For example,
if you specify 5 as your REVOKE number, a user will be revoked upon
three consecutive incorrect passwords followed by three consecutive
incorrect password phrases.
NOREVOKE
Specifies that RACF ignores the number of consecutive unsuccessful attempts to access the system using an incorrect password or password phrase.

RULEn | NORULEn | NORULES

Tip
You might find the ISPF panels easier to use for entering password rules.

RULEn (LENGTH (m1:m2) content-keyword (position))
Specifies an individual syntax rule for new passwords that users specify at logon, on JCL job cards, or on the PASSWORD command. Also applies to passwords specified on the ALTUSER commands that have the NOEXPIRED operand. Eight syntax rules are allowed. Therefore, for the RULEn suboperand, the value of n is 1–8.

These syntax rules do not apply to:
- Password phrases
- Logon passwords that are currently in effect for a user
- Logon passwords specified on the ADDUSER command
- Logon passwords specified on the ALTUSER command with the PASSWORD operand and with the EXPIRED operand either specified or defaulted
- Default passwords set by the PASSWORD USER(userid) command, which are set to the user’s default group name.

If multiple rules are defined, a password that passes at least one rule is accepted.

Restriction: Changes to password syntax rules will not force users to immediately change their passwords. RACF does not apply new password rules to users until users change their passwords—either voluntarily or at password expiration.

LENGTH(m1:m2)
Specifies the minimum and maximum password lengths to which this particular rule applies (m2 must be greater than or equal to m1). Because RACF allows passwords no longer than 8 alphanumeric characters, the value for m2 must be less than or equal to 8. If you omit the m2 value, the rule applies to a password of one length only.

content-keyword(position)
Specifies the syntax rules for the positions indicated by the LENGTH suboperand. Rules specifying mixed-case characters should only be set when the MIXEDCASE option is in effect. New passwords will not match these rules when mixed-case passwords are not supported, either because the MIXEDCASE option is not in effect or because an application is used that does not support mixed-case passwords. The possible values for content-keyword are:
ALPHA
Includes uppercase alphabetic characters and the national
characters # (X’7B’), $ (X’5B’), and @ (X’7C’)

ALPHANUM
Includes the ALPHA characters—uppercase alphabetic
characters and the national characters # (X’7B’), $ (X’5B’), and @
(X’7C’)—and NUMERIC characters.

If the password syntax rule requires only one ALPHANUM
character, passwords must contain either one ALPHA character
or one NUMERIC character.

If the password syntax rule requires two or more ALPHANUM
characters, passwords must contain at least one ALPHA
character and at least one NUMERIC character.

VOWEL
Includes uppercase vowel characters, namely A, E, I, O, and U

NOVOWEL
Includes characters that are not vowels, such as
- Uppercase alphabetic characters that are consonants, not
vowels
- National characters
- Numeric characters

CONSONANT
Includes uppercase non-vowel characters

NUMERIC
Includes numeric characters

NATIONAL
Includes the national characters # (X’7B’), $ (X’5B’), and @
(X’7C’)

MIXEDCONSONANT
Includes uppercase and lowercase non-vowel characters

MIXEDVOWEL
Includes the uppercase and lowercase vowel characters, A, E, I,
O, U, and a, e, i, o, u

MIXEDNUM
Includes all characters of the following three types of
MIXEDNUM characters:
1. ALPHA characters—includes uppercase alphabetic
   characters and the national characters # (X’7B’), $ (X’5B’),
   and @ (X’7C’)
2. Lowercase alphabetic characters
3. NUMERIC characters.

If the password syntax rule requires only one MIXEDNUM
character, passwords must contain at least one character of any
one of the three MIXEDNUM character types.

If the password syntax rule requires two MIXEDNUM characters,
passwords must contain two characters of different MIXEDNUM
character types, in one of the following valid combinations:
- An ALPHA character and a lowercase alphabetic
SETROPTS

- An ALPHA character and a NUMERIC character
- A lowercase alphabetic character and a NUMERIC character.

If the password syntax rule requires three or more MIXEDNUM characters, passwords must contain three or more MIXEDNUM characters including at least one character of each MIXEDNUM character type.

If the values in the content-keywords do not define every position specified by the LENGTH value, the undefined positions can consist of any combination of alphanumeric characters.

Each content-keyword is followed by a position (in the form of $k$, not greater than 8), list of positions (form of $k1,k2,k3...$ in any order), or a range (form of $k4:k5$, where $k5$ must be greater than or equal to $k4$).

Example:
RULE1(LENGTH(8) CONSONANT(1,3,5:8) NUMERIC(2,4))

Result:
Syntax RULE1 applies to passwords eight characters in length with consonants in positions 1, 3, 5, 6, 7, and 8 and numbers in positions 2 and 4. The password B2D2GGDD obeys RULE1, and C3PIBOLO does not.

Example:
RULE2(LENGTH(6) NATIONAL(3) MIXEDNUM(4:6))

Result:
Syntax RULE2 applies to passwords 6 characters in length with a national character in position 3 and requires an uppercase alphabetic, a lowercase alphabetic, and a numeric in positions 4, 5, and 6. The password AB@1tD obeys RULE2.

NORULE$n$
Specifies that RACF is to delete the particular rule identified by $n$.

NORULES
Specifies that RACF is to delete all password syntax rules established by the installation.

NORULES is in effect when RACF is using a newly initialized database.

WARNING | NOWARNING

WARNING(days-before-password-expires)
Specifies the number of days (1–255) before a password or password phrase expires, indicating that RACF is to issue a warning message to the TSO user or to the job log of a batch job that specified the expiring password or password phrase.

If you specify a WARNING value that exceeds the INTERVAL value, a warning message is issued at each logon. If you do not want the warning with each logon, specify a value for WARNING that is less than the value you specify for INTERVAL. If you specify WARNING, INITSTATS must be in effect.

NOWARNING
Specifies that RACF is not to issue the warning message for expiring passwords or password phrases.
NOWARNING is in effect when RACF is using a newly initialized database.

PREFIX | NOPREFIX
PREFIX
Activates RACF protection for data sets that have single-qualifier names, and specifies the 1–8 character prefix to be used as the high-level qualifier in the internal form of the names. The variable prefix should be a predefined group name, and it must not be the high-level qualifier of any actual data sets in the system.

NOPREFIX
Deactivates RACF protection for data sets that have single-level names.

When EGN is active and NOPREFIX is in effect, a data set can be protected with a generic profile of the form ABC.**, where ABC equals the data set name.

NOPREFIX is in effect when RACF is using a newly initialized database.

PROTECTALL | NOPROTECTALL
PROTECTALL(FAILURES | WARNING)
Activates PROTECTALL processing. When PROTECTALL processing is active, the system automatically rejects any request to create or access a data set that is not RACF-protected. This processing includes DASD data sets, tape data sets, catalogs, and GDG basenames. Temporary data sets that comply with standard MVS temporary data set naming conventions are excluded from PROTECTALL processing.

Note that PROTECTALL requires all data sets to be RACF-protected. This includes tape data sets if your installation specifies the TAPEDSN operand on the SETROPTS command.

In order for PROTECTALL to work effectively, you must specify GENERIC to activate generic profile checking. Otherwise, RACF would allow users to create or access only data sets protected by discrete profiles. If your installation uses nonstandard names for temporary data sets, you must also predefine entries in the global access checking table that allow these data sets to be created and accessed.

The WARNING suboperand enables you to specify a warning message to the requestor in place of rejecting the request.

FAILURES
Specifies that RACF is to reject any request to create or access a data set that is not RACF-protected.

The default value is FAILURES.

If PROTECTALL(FAILURES) is in effect and a user with the SPECIAL attribute requests access to an unprotected data set, RACF accepts the request, audits the event, and issues a PROTECTALL warning message.

If PROTECTALL(FAILURES) is in effect and a trusted started task requests access to an unprotected data set, RACF accepts the request, audits the event, and no warning message is issued.
SETROPTS

If PROTECTALL(FAILURES) is in effect and a privileged started task requests access to an unprotected data set, RACF accepts the request, the event is not audited, and no warning message is issued.

**WARNING**

Specifies that when a user requests creation of, or access to, a data set that is not RACF-protected, RACF is to allow the request but issue warning messages to the user and the security administrator.

**NOPROTECTALL**

Specifies that a user can create or access a data set that is not protected by a profile.

NOPROTECTALL is in effect when RACF is using a newly initialized database.

**RACLIST | NORACLIST**

**RACLIST(class-name ...)**

Activates the sharing of in-storage profiles, both generic and discrete, for the classes specified. Also see GENLIST operand.

Activate this function to improve the performance of resource access checking for a general resource class. With the profiles for the class in storage, RACF requires no database I/O when making an access decision.

A valid class-name is any member class for which the class descriptor table allows or requires RA CLIST processing. Grouping classes are not valid, except for RACFVARS and NODES. If class-name is valid, not only the specified class-name, but all classes that share the same POSIT are processed. If some classes sharing the same POSIT do not allow RA CLIST processing, those classes are skipped.

Only active classes are RA CLISTed. Be sure to activate each class you want to RA CLIST. For example:

```
SETROPTS RA CLIST(DIGTCERT) CLASSACT(DIGTCERT)
```

If REFRESH is also specified, member classes for which the class descriptor table does not allow RA CLIST processing are also valid because the SETROPTS RA CLIST(class-name) REFRESH command refreshes classes that were RA CLISTed by RACR OUTE REQUEST=LIST,GLOBAL=YES or SETROPTS RA CLIST. Likewise, classes for which SETROPTS GENLIST was specified are also valid.

You cannot SETROPTS RA CLIST and SETROPTS GENLIST for the same general resource class.

**Rule:** If the following supplied classes are active, you must issue the SETROPTS RA CLIST command to share them:

- APPC S ERV
- APPC T P
- CSF KEYS
- CSF S ERV
- DE V ICE S
- DIGTN M AP
- FIELD
- ID ID M AP
- NODE S
- OPERCM DS
- PROPCNTL
- PTKTDATA
- RACF VARS
- SECLABEL
- SERVA U TH
- SERVAUTH
- SY S MVIE W
- UN IXPRIV
- STARTED
- VTAMAPPL

In-storage profiles for the following supplied classes can be optionally shared by using SETROPTS RA CLIST.
Important: For each class marked with an asterisk (*), you might incur performance degradation or missing function if you do not issue the SETROPTS RACLIST command when you define profiles in the class and activate it. For important details about each class, see [z/OS Security Server RACF Security Administrator's Guide](https://www.ibm.com) (for classes used for RACF functions) or the appropriate program documentation.

If you have, or are considering, authorizing a large number of users for a resource in a class that can be processed to an in-storage profile using the SETROPTS RACLIST command, you must consider the number of entries in the access list, because RACLST processing merges profiles and the access lists of each profile. The combined number of access-list entries may cause the profile to become too large to be processed, and RACLST processing may fail. See [z/OS Security Server RACF Security Administrator's Guide](https://www.ibm.com) for more information about limiting the size of access lists and profile sizes.

Notes:

1. When you activate RACLST processing for a class, RACF copies both discrete and generic profiles for that class into a data space.

2. When the RACGLIST class is active and class-name profiles have been specified in the RACGLIST class, SETROPTS RACLIST(class-name) stores the RACLSTed results from the data space in the RACGLIST classname_nnnnn profiles on the RACF database, enabling all systems sharing the database to access the same level of profile information.

   For example if you issue the commands:

   ```
   SETR CLASSACT(RACGLIST)
   RDEFINE RACGLIST TERMINAL
   SETROPTS RACLST(TERMINAL)
   ```

   Then either when you issue:

   ```
   SETROPTS RACLST(TERMINAL)
   ```

   or at the next IPL, if the TERMINAL class was RACLSTed before the RACGLIST class was activated, RACF creates RACGLIST TERMINAL_00001, RACGLIST TERMINAL_00002, and so on, to hold the results of the SETROPTS RACLST processing.

   The profiles are available to all authorized users, thereby eliminating the need for RACF to retrieve a profile each time a user requests access to a resource protected by that profile. Thus, when you activate this function, you reduce processing overhead.

   The SETROPTS RACLST(class-name) command overrides a RACROUTE REQUEST=LIST,GLOBAL=YES request for the same class. The data space and RACGLIST classname_nnnnn profiles, if any,
SETROPTS

are refreshed by the SETROPTS RACLIST. SETROPTS LIST output
will list the class in the SETR RACLIST CLASSES = line rather than the
GLOBAL=YES RACLIST ONLY = line.

3. If you specify RACLST with REFRESH, RACF rebuilds the discrete and
generic profiles for the class and places them in the new data space. If
the RACGLIST class is active and contains a profile for class-name, the
classname_nnnnn profiles for the class are also rebuilt, or are created if
they had not been built previously.

SETROPTS RACLST(class-name) REFRESH can also be used to
refresh classes RACLSTed by RACROUTE
REQUEST=LIST,GLOBAL=YES, as well as classes that are
RACLSTed. It refreshes the class, but has no effect on SETROPTS
LIST output. If the class was processed using SETROPTS RACLST
solely by RACROUTE
REQUEST=LIST,ENVIR=CREATE,GLOBAL=YES, the class are listed in
the GLOBAL=YES RACLIST ONLY = line. Regardless of whether the class
was RACLSTed by GLOBAL=YES, if it was RACLSTed by SETROPTS
RACLST (classname) then the class is listed only in the SETR RACLIST
CLASSES = line.

SETROPTS RACLST(classname) REFRESH can also be issued to
create the RACGLIST profiles for the class, even if the class were not
RACLSTed by either RACROUTE REQUEST=LIST,GLOBAL=YES or
by SETROPTS RACLST. Then the first RACROUTE
REQUEST=LIST,GLOBAL=YES uses the RACLIST profiles to build the
RACLST data space, rather than accessing the database for each
individual discrete and generic profile.

While the rebuild is in progress, RACF continues to use the old
in-storage profiles for authorization requests until the new ones are
created. When all systems have completed rebuilding the local data
spaces, the coordinator signals the members of the data sharing group
to discard the old ones, and to begin using the new one.

4. When RACF is enabled for sysplex communication, RACF propagates a
SETROPTS RACLST(class-name) or SETROPTS
RACLST(class-name) REFRESH command issued from any one
system (coordinator) to other systems in the data sharing group (peers)
if the command is successful on the system on which it was entered. If
the RACGLIST classname_nnnnn profiles were built for the class, peer
members of the sysplex use the results to build the RACLST data
space on their system, but do not rebuild the RACGLIST profiles.

If a refresh is being done, RACF continues to use the old in-storage
profiles for authorization requests until the new ones are
created. When all systems have completed rebuilding the local data
spaces, the coordinator signals the members of the data sharing group to discard
the old ones, and to begin using the new one.

If RACF is not enabled for sysplex communication, you must issue the
SETROPTS RACLST(class-name) command and the SETROPTS
RACLST(class-name) REFRESH command on each system sharing
the database.

5. When you activate RA CLIST processing for the CDT class, the dynamic
class descriptor table is built in a dataspace instead of in-storage
profiles. The information in the dataspace is not used for authorization
checking. If authorization checking using RACROUTE
REQUEST=FASTAUTH is required for the CDT class, you must use
RACROUTE REQUEST=LIST,GLOBAL=NO to locally RA CLIST the
CDT class profiles. Alternatively, RACROUTE REQUEST=AUTH may be used for the CDT class, and RACF will use CDT profiles in the RACF database for authorization checking. For more information on the dynamic CDT, see z/OS Security Server RACF Security Administrator’s Guide.

NORACLIST(class-name ...)

Deactivates the sharing of in-storage profiles, both generic and discrete, for the classes specified. Also see the NOGENLIST operand.

When you specify NORACLIST, RACF deletes the data space containing the generic and discrete profiles for the specified classes. The data space might have been created by specifying the class with either a SETROPTS RACLST command or a RACROUTE REQUEST=LIST,GLOBAL=YES request. In the latter case, all applications that issued a RACROUTE REQUEST=LIST,ENVIR=CREATE,GLOBAL=YES for the class should issue a RACROUTE REQUEST=LIST,ENVIR=DELETE before a SETROPTS NORACLST is issued that processes the class. The SETROPTS NORACLST should be used to delete the data space only after all applications have relinquished their access to it.

For both the SETROPTS RACLST and RACROUTE REQUEST=LIST,GLOBAL=YES cases, if RACLIST classname_nnnnn profiles exist for the class, they are deleted. Even if the class was not RACLSTed, SETROPTS NORACLST can be used to delete these profiles. In all cases, the RACGLIST classname profile remains.

A valid class-name is any member class in the class descriptor table. Grouping classes are not valid, except for RACFVARS and NODES. If class-name is valid, not only the specified class but all classes that share the same POSIT are processed. For a list of general resource classes defined in the class descriptor table supplied by IBM, see Appendix B, “Supplied RACF resource classes,” on page 659.

Because SETROPTS NORACLST, like SETROPTS RACLST REFRESH, operates on classes that are RACLSTed by RACROUTE REQUEST=LIST,GLOBAL=YES, or SETROPTS RACLST, member classes in the class descriptor table that do not allow RACLST processing are now valid classes for the command. Both these conditions are still invalid for SETROPTS RACLST.

When RACF is enabled for sysplex communication, RACF propagates the SETROPTS NORACLST command to other systems in the data sharing group, if the command was successful on the system in which it was entered. If RACF is not enabled for sysplex communication, you must issue the SETROPTS NORACLST command on each system sharing the database.

NORACLST is in effect for all classes defined in the class descriptor table when RACF is using a newly initialized database.

When SETROPTS NORACLST(CDT) is issued, the dataspace containing the dynamic class descriptor table is deactivated, but not deleted. The dataspace remains until the system is restarted. For more information on the dynamic CDT, see z/OS Security Server RACF Security Administrator’s Guide.

REALDSN | NOREALDSN

REALDSN

Specifies that RACF is to record, in any SMF log records and operator...
SETROPTS

messages, the real data set name (not the naming-conventions name) used on the data set commands and during resource access checking and resource definition.

NOREALDSN
Specifies that RACF is to record, in any SMF log records and operator messages, the data set names modified according to RACF naming conventions.

NOREALDSN is in effect when RACF is using a newly initialized database.

REFRESH
Refreshes the in-storage generic profiles when specified with GENERIC, GLOBAL or RACLIST, or the in-storage program control tables when specified with WHEN(PROGRAM).

RETPD(*nnnnn)
Specifies the default RACF security retention period for tape data sets, where *nnnnn is a 1-5 digit number in the range of 0 through 65533 or 99999 to indicate a data set that never expires. The security retention period is the number of days that RACF protection is to remain in effect for a tape data set; RACF stores the value in the tape data set profile.

If you specify RETPD, you must also specify TAPEDSN to activate tape data set protection. If you omit TAPEDSN, RACF records the value you specify for security retention period in the list of RACF options. However, without tape data set protection activated, this value is meaningless.

If you specify RETPD and TAPEDSN, the value you specify for security retention period is the default for your installation; RACF places the value in each tape data set profile unless the user specifies one of the following:
- An EXPDT in the JCL other than the current date
- An RETPD other than 0 on the ADDSD command.

If you specify TAPEDSN and do not specify RETPD, RACF uses a value of 0 for the default security retention period.

RVARYPW([SWITCH(sw-switch-pw)] [STATUS(status-switch-pw)])
Specifies the passwords that the operator is to use to respond to requests to approve RVARY command processing, where sw-switch-pw is the response to a request to switch RACF databases or change the operating mode of RACF, and status-switch-pw is the response to a request to change RACF or database status from ACTIVE to INACTIVE or from INACTIVE to ACTIVE. You can specify different passwords for each response. Note that NO is not a valid password for either SWITCH or STATUS.

When RACF is using a newly initialized database, the switch password and the status password are both set to YES.

SAUDIT | NOSAUDIT
Specifies whether RACF is to log RACF commands issued by users with the SPECIAL or group-SPECIAL attribute. You must have the AUDITOR attribute to specify these operands.

SAUDIT
Specifies that RACF is to log RACF commands (except LISTDSD, LISTGRP, LISTUSER, RLIST, and SEARCH) issued by users who either had the SPECIAL attribute or who gained authority to issue the command through the group-SPECIAL attribute.

SAUDIT is in effect when RACF is using a newly initialized database.
**SETROPTS**

**NOSAUDIT**
Specifies that RACF is not to log the commands issued by users with the SPECIAL or group-SPECIAL attribute.

**SECLABELAUDIT | NOSECLABELAUDIT**
You must have the AUDITOR attribute to specify these options.

**SECLABELAUDIT**
Specifies that the SECLABEL profile’s auditing options are to be used in addition to the auditing options specified for the user or resource. This additional auditing occurs whenever an attempt is made to access or define a resource protected by a profile, FSP, or ISP that has a security label specified, or when a user running with a security label attempts to access or define a resource.

The SECLABEL profile requires SETROPTS RACLIST processing. If SECLABEL profile audit options are not specified, SECLABEL auditing is not done.

For more information, refer to [*z/OS Security Server RACF Auditor’s Guide*](#).

**NOSECLABELAUDIT**
Disables auditing by SECLABEL.

NOSECLABELAUDIT is in effect when RACF is using a newly initialized database.

**SECLABELCONTROL | NOSECLABELCONTROL**

**SECLABELCONTROL**
Limits the users who can specify the SECLABEL operand on RACF commands. Those allowed to specify the operand are:
- Users with the SPECIAL attribute can specify the SECLABEL operand on any RACF command.
- Users with the group-SPECIAL attribute can specify the SECLABEL operand on the ADDUSER and ALTUSER commands when adding a user to a group within their scope of control (provided the group-SPECIAL is permitted to the SECLABEL).

**NOSECLABELCONTROL**
Allows any user to change the SECLABEL field in a profile, as long as the user has at least READ access authority to the associated SECLABEL profile.

NOSECLABELCONTROL is in effect when RACF is using a newly initialized database.

**SECLBYSYSTEM | NOSECLBYSYSTEM**

**SECLBYSYSTEM**
Specifies that security labels can be activated on a system image basis. When SECLBYSYSTEM is active, the SMF ID values specified in the member list of the profiles in the SECLABEL class will determine whether or not the security label is valid for each system. Security labels that are not valid for a system are considered inactive and cannot be used or listed by users without SPECIAL or AUDITOR on that system. After activating SECLBYSYSTEM, SETR RACLIST(SECLABEL) REFRESH must be issued to complete the activation of security labels by system. This option cannot be activated if the SECLABEL class is not active.

**NOSECLBYSYSTEM**
Specifies that security labels are not activated on a system image basis.
SETROPTS

NOSECCLBSYSTEM is in effect when RACF is using a newly initialized database.

**SECLEVELAUDIT | NOSECLEVELAUDIT**

You must have the AUDITOR attribute to specify these operands.

**SECLEVELAUDIT (security-level)**

Activates auditing of access attempts to all RACF-protected resources based on the specified installation-defined security level. RACF audits all access attempts for the specified security level and higher.

You can specify only a security level name defined by your installation as a SECLEVEL profile in the SECDATA class. (For information on defining security levels, see the description of the RDEFINE and RALTER commands.)

**NOSECLEVELAUDIT**

Deactivates auditing of access attempts to RACF-protected resources based on a security level.

NOSECLEVELAUDIT is in effect when RACF is using a newly initialized database.

**SESSIONINTERVAL | NOSESSIONINTERVAL**

**SESSIONINTERVAL(n)**

Sets the maximum value that can be specified by RDEFINE or RALTER for session key intervals. This $n$ value must be a number in the range of 1–32767 (inclusive).

The SESSIONINTERVAL value after RACF data set initialization is 30. This value is used for:

1. A default if SESSION is specified without INTERVAL on RDEFINE when defining an APPCLU class profile.
2. An upper limit if INTERVAL is specified on RDEFINE or RALTER for APPCLU class profiles.

**NOSESSIONINTERVAL**

Disables the global limit on the number of days before a session key expires. The internal value is set to zero.

**STATISTICS | NOSTATISTICS**

Use these operands to cause RACF to record or not record statistical information for the specified class name. The valid class names are DATASET and those classes defined in the class descriptor table. For a list of general resource classes defined in the class descriptor table supplied by IBM, see Appendix B, “Supplied RACF resource classes,” on page 659.

**Note:** If you activate or deactivate statistics processing for a class, all other classes in the class descriptor table with the same POSIT number are also be activated or deactivated. If, for instance, you activate statistics processing for the TIMS class, statistics processing is activated for classes AIMS and GIMS because they share POSIT number 4.

For more information on sharing a POSIT value, see the POSIT keyword of the RDEFINE command.

**STATISTICS(class-name ... | *)**

Specifies that RACF is to record statistical information for class-name.
If you specify an asterisk (*), you activate the recording of statistical information for the DATASET class and all classes defined in the class descriptor table.

When RACF is using a newly initialized database, the recording of class statistics is turned off. Because statistics recording has an impact on system performance, it is recommended that you do not activate this option for any class until your installation evaluates the need to use it versus the potential performance impact. For more information, see z/OS Security Server RACF System Programmer's Guide.

NOSTATISTICS(class-name ... | *)
Specifies the names of the classes to be deleted from those previously defined to have statistical information recorded.

If you specify an asterisk (*), you deactivate the recording of statistical information for the DATASET class and all classes defined in the class descriptor table.

TAPEDSN | NOTAPEDSN

TAPEDSN
Activates tape data set protection. When tape data set protection is in effect, RACF can protect individual tape data sets as well as tape volumes.

If you activate tape data set protection, you should also activate the TAPEVOL class. If you do not also activate TAPEVOL, RACF does not check the retention period before it deletes a tape data set, and you must provide your own protection for tape data sets that reside on a volume that contains more than one data set.

Before you activate tape data set protection, see z/OS Security Server RACF Security Administrator's Guide for a complete description of the relationship between TAPEDSN and activating the TAPEVOL class.

NOTAPEDSN
Deactivates tape data set protection. When NOTAPEDSN is in effect, RACF cannot protect individual tape data sets, though it can protect tape volumes.

NOTAPEDSN is in effect when RACF is using a newly initialized database.

TERMINAL(READ | NONE)
Is used to set the universal access authority (UACC) associated with undefined terminals. If you specify TERMINAL but do not specify READ or NONE, the system prompts you for a value.

WHEN | NOWHEN

WHEN(PROGRAM)
Activates RACF program control, which includes both access control to load modules and program access to data sets.

To set up access control to load modules, you must identify your controlled programs by creating a profile for each in the PROGRAM class. To set up program access to data sets, you must add a conditional access list to the profile of each program-accessed data set. Then, when program control is active, RACF ensures that each controlled load module is executed only by callers with the defined authority. RACF also ensures that each program-accessed data set is opened only by users who are listed in the conditional access list with the proper authority and who are executing the program specified in the conditional access list entry.
When RACF is enabled for sysplex communication, the SETROPTS WHEN(PROGRAM) command and the SETROPTS WHEN(PROGRAM) REFRESH command are propagated to other members of the data sharing group if the command was successful on the system on which it was entered. When RACF is not enabled for sysplex communication, you must issue the SETROPTS WHEN(PROGRAM) command and the SETROPTS WHEN(PROGRAM) REFRESH command on each system sharing the database.

For more information about program control, see z/OS Security Server RACF Security Administrator's Guide.

**Note:** The PROGRAM class does not have to be active.

**NOWHEN(PROGRAM)**

Specifies that RACF program control is not to be active.

NOWHEN(PROGRAM) is in effect when RACF is using a newly initialized database.

**Examples**

**Table 94. SETROPTS examples**

**Example 1**

**Operation**

User FRG34 wants to establish logging options that causes RACF to log all activity in the USER and GROUP classes, log the activities of users with the SPECIAL and group-SPECIAL attributes, log all accesses allowed only because the user has the OPERATIONS or group-OPERATIONS attribute, log all command violations, and audit all attempts to access RACF-protected resources based on the installation-defined security level SECRET.

**Known**

User FRG34 has the AUDITOR attribute. SECRET is defined as a SECLEVEL profile in the SECDATA class.

User FRG34 wants to issue this command as a RACF TSO command.

**Command**

SETROPTS AUDIT(USER GROUP) OPERAUDIT SECLEVELAUDIT(SECRET)

**Defaults**

SAUDIT CMDVIOL

**Example 2**

**Operation**

User RVU03 wants to establish a set of syntax rules for passwords that obey the following rules:

- The minimum password length is 4 characters
- Four character passwords must have at least one numeric and one alphabetic character
- Five character passwords must contain at least one numeric character or be completely alphabetic
- Passwords of 6 or more characters consist of any combination of alphabetic and numeric characters.

**Known**

User RVU03 has the SPECIAL attribute.

User RVU03 wants to issue this command as a RACF TSO command.

**Command**

SETROPTS PASSWORD(RULE1(LENGTH(4:5) ALPHANUM(1:5)) RULE2(LENGTH(5) ALPHA(1:5))) RULE3(LENGTH(6:8) ALPHANUM(1:8)) RULE4(LENGTH(6:8) NUMERIC(1:8)) RULE5(LENGTH(6:8) ALPHA(1:8)))

**Defaults**

None.

**Example 3**

**Operation**

User ADM1 wants to display the RACF options currently in effect. MVS and VM systems share the RACF database.

**Known**

User ADM1 has the SPECIAL and AUDITOR attributes.

User ADM1 wants to issue this command as a RACF TSO command.

**Command**

SETROPTS LIST

**Defaults**

None.

**Output**

See Figure 74 on page 626 for a sample listing.
SETROPTS

Table 94. SETROPTS examples (continued)

Example 4  
Operation User RVU02 wants to establish system-wide options for an installation. The installation requires tape data set protection and tape volume protection, and the maximum change interval is to be 60 days. The default RACF security retention period for tape data sets is to be 360 days.

Known User RVU02 has the SPECIAL attribute.

User RVU02 wants to issue this command as a RACF TSO command.

Command SETROPTS PASSWORD(INTERVAL(60)) CLASSACT(TAPEVOL) TAPEDSN RETPD(360)

Defaults None.

Example 5  
Operation User ADM1 wants to enable the generic profile checking facility for the DATASET class.

Known User ADM1 has the SPECIAL attribute.

User ADM1 wants to issue this command as a RACF TSO command.

Command SETROPTS GENERIC(DATASET)

Defaults None.

Example 6  
Operation User ADM1 wants to activate global access checking for the DATASET class.

Known User ADM1 has the SPECIAL attribute.

User ADM1 wants to issue this command as a RACF TSO command.

Command SETROPTS GLOBAL(DATASET)

Defaults None.

Example 7  
Operation User ADM1 wants to activate erase-on-scratch processing for all resources with a security level of CONFIDENTIAL or higher and set the SWITCH and STATUS passwords for the RVARY command.

Known User ADM1 has the SPECIAL attribute. The CONFIDENTIAL security level name is known to RACF.

User ADM1 wants to issue this command as a RACF TSO command.

Command SETROPTS ERASE(SECLEVEL(CONFIDENTIAL)) RVARYPW(SWITCH(LINUS) STATUS(LUCY))

Defaults None.

Example 8  
Operation The RACF system administrator wants to activate installation defaults for the primary and secondary national languages. The primary language is Japanese and the secondary language is Canadian French.

Known The system administrator has the SPECIAL attribute. The MVS message service is not active. The 3-character language code for Japanese is JPN. The language code for Canadian French is FRC.

The system administrator wants to issue this command as a RACF TSO command.

Command SETROPTS LANGUAGE(PRIMARY(JPN) SECONDARY(FRC))

Defaults None.
The second line of this display, `ATTRIBUTES =`, refers to global RACF attributes in effect. These attributes can be set only with the `SETROPTS` command. They are different from, and should not be confused with, the RACF user attributes.
### PASSWORD PROCESSING OPTIONS:
- **Password Change Interval is 254 Days.**
- **Password Minimum Change Interval is 2 Days.**
- **Mixed Case Password Support is in effect.**
- **13 Generations of Previous Passwords Being Maintained.**
- **After 4 Consecutive Unsuccessful Password Attempts, a Userid Will Be Revoked.**
- **Password Expiration Warning Level is 186 Days.**

### INSTALLATION PASSWORD SYNTAX RULES:

<table>
<thead>
<tr>
<th>Rule</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Length(4:5) LLLLL</td>
</tr>
<tr>
<td>2</td>
<td>Length(5) AAAAA</td>
</tr>
<tr>
<td>3</td>
<td>Length(6:8) LLLLLLL</td>
</tr>
<tr>
<td>4</td>
<td>Length(6:8) NNNNNNN</td>
</tr>
<tr>
<td>5</td>
<td>Length(6:8) AAAAAAAA</td>
</tr>
</tbody>
</table>

**Legend:**
- A - Alpha
- C - Consonant
- L - Alphanumeric
- N - Numeric
- V - Vowel
- W - No Vowel
- * - Anything
- c - Mixed Consonant
- m - Mixed Numeric
- v - Mixed Vowel
- $ - National

### DEFAULT RVARIES:
- **Password is in effect for the switch function.**
- **Password is in effect for the status function.**

### SECLABELS:
- **Audit is in effect.**
- **Control is in effect.**
- **Generic Owner Only is in effect.**
- **Compatibility Mode is in effect.**
- **Multi-Level Quiet is in effect.**
- **Multi-Level Stable is in effect.**
- **No Write-Down is in effect.**
- **Current options:**
  - "MLS Warning" option is in effect.
  - "Multi-Level Secure is in effect."
  - "Multi-Level Active is in effect."

### LANGUAGES:
- **Primary Language Default:** ENU / AMERICAN
- **Secondary Language Default:** ENU / AMERICAN

---

**Note:** The language name (in this example, AMERICAN) only appears if the MVS message service is active.

---

*Figure 74. Output for SETROPTS LIST (Part 2 of 2)*
SIGNOFF

SIGNOFF (Sign off sessions)

Background

Persistent verification allows users to sign on to a partner LU (logical unit) and have their authority persist. In other words, once a user has signed on, a password is not required for subsequent signon attempts.

APPC/MVS invokes RACF to create and maintain a list called the signed-on-from list. If persistent verification is being used, the signed-on-from list consists of the users currently signed on with Persistent Verification authority.

Purpose

The RACF SIGNOFF operator command removes user entries from the signed-on-from list. Entries in the signed-on-from list are selected by the SIGNOFF command using the following information:

- User ID
- Group
- APPL (the local LU name)
- POE (the partner LU name from which the user is signed on)

The SIGNOFF command has operands which correspond to the items above. You can use these operands to select which user entries to remove from the signed-on-from list.

To determine which user entries are signed off by issuing a particular SIGNOFF command, issue a DISPLAY command with corresponding selection criteria.

Issuing options

The following table identifies the eligible options for issuing the SIGNOFF command:

<table>
<thead>
<tr>
<th>As a RACF TSO command?</th>
<th>As a RACF operator command?</th>
<th>With command direction?</th>
<th>With automatic command direction?</th>
<th>From the RACF parameter library?</th>
</tr>
</thead>
<tbody>
<tr>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>

For information on issuing this command as a RACF operator command, see "Rules for entering RACF operator commands" on page 21. You must be logged on to the console to issue this command as a RACF operator command.

Related commands

Use the DISPLAY operator command to view the signed-on-from list.

Authorization required

When issuing the SIGNOFF command as a RACF operator command, you might require sufficient authority to the proper resource in the OPERCMDS class. See z/OS Security Server RACF Security Administrator's Guide for further information.
Syntax

For the key to the symbols used in the command syntax diagrams, see "Syntax of RACF commands and operands" on page 9. The complete syntax of the SIGNOFF command is:

```
subsystem-prefix SIGNOFF
  APPL(local-luname | *)
  POE(partner-luname | *)
  USER(userid-name | *)
  [ GROUP(group-name | *) ]
  [ SECLABEL(security-label | *) ]
```

For information on issuing this command as a RACF operator command, see "Rules for entering RACF operator commands" on page 21. You must be logged on to the console to issue this command as a RACF operator command.

Parameters

**subsystem-prefix**

Specifies that the RACF subsystem is the processing environment of the command. The subsystem prefix can be either the installation-defined prefix for RACF (1–8 characters) or, if no prefix has been defined, the RACF subsystem name followed by a blank. If the command prefix was registered with CPF, you can use the MVS command D OPDATA to display it or you can contact your RACF security administrator.

Only specify the subsystem prefix when issuing this command as a RACF operator command. The subsystem prefix is required when issuing RACF operator commands.

The operands listed below allow the operator to specify the user entries to be signed off. The **APPL**, **POE** and **USER** operands are required to uniquely identify a user entry to be signed off. The **GROUP** operand is optional and defaults to a **group-name** consisting of blanks.

**APPL(local-luname | *)**

This is a required operand. The **local-luname** is a 1–8 character name of the local LU to be searched for. An asterisk can occupy the last position of the **local-luname** in order to provide a partial generic selection capability. A character string consisting of a single asterisk is permitted as a full generic that matches any APPL name in the signed-on-from list.

**POE(partner-luname | *)**

This is a required operand. The **partner-luname** is a 1–7 character name of the partner LU to be searched for. A **partner-luname** consisting of a single asterisk is permitted as a full generic that matches any POE name in the signed-on-from list.

**USER(userid-name | *)**

This is a required operand. The **userid-name** is a 1–8 character specification which represents the RACF user ID to be searched for. A character string consisting of a single asterisk is permitted as a full generic that matches any user ID in the signed-on-from list.

**GROUP(group-name | *)**

This is an optional operand. The **group-name** is a 1–8 character name which represents the RACF group to be searched for. A character string consisting of
SIGNOFF

A single asterisk is also permitted as a full generic which matches any group-name in the signed-on-from list. If this operand is not specified, the default value is a group-name consisting of blanks.

Note that entries in the signed-on-from list might not always be added to that list with a group-name value. Such entries have group-names consisting of blanks.

SECLABEL(security-label | *)
This is an optional operand. The security-label is a 1–8 character name that represents the RACF security label to be searched for. This operand is currently ignored.

Examples

Table 95. SIGNOFF examples

Example 1
Operation Sign off a user from a local/partner LU pair.
Known The local LU is locallu, the partner LU is prtnrlu5, and the userid-name is jim. The RACF subsystem prefix is @.
Command @signoff user(jim),appl(locallu),poe(prtnrlu5)
Defaults A group-name consisting of blank characters.

Example 2
Operation Sign off all of the users from a local/partner LU pair.
Known The local LU is locallu, the partner LU is prtnrlu5, and the RACF subsystem prefix is @.
Command @signoff appl(locallu),poe(prtnrlu5),user(*)
Defaults A group-name consisting of blank characters.

Example 3
Operation Sign off a user from all the local/partner LU pairs to which that user is signed on.
Known The userid-name is Kurt, and the RACF subsystem prefix is @.
Command @signoff appl(*),poe(*),group(*),user(jim)
Defaults None.

Example 4
Operation Sign off all users from all the partner LUs of a particular local LU.
Known The local LU is locallu, the RACF subsystem prefix is @.
Command @signoff appl(locallu),poe(*),user(*),group(*)
Defaults None.

Example 5
Operation Sign off all users of a particular group from a particular local LU.
Known The local LU is locallu, the group is grp1, and the RACF subsystem prefix is @.
Command @signoff appl(locallu),poe(*),user(*),group(grp1)
Defaults None.
STOP (Shutdown RRSF)

Purpose

Use the STOP command to allow the MVS operator to stop the RACF subsystem address space if restarting a subtask is not sufficient to recover from a failure. This command shuts down the RACF subsystem address space and prevents the loss of any outstanding requests that are waiting for completion.

The STOP command can also be used to stop the RACF subsystem address space before an IPL.

Guidelines:

- If you are directing work to the RACF subsystem, stop the RACF subsystem address space before IPLing to prevent the loss of outstanding requests.
- If you are using automatic direction or password synchronization, do not stop the address space except immediately before an IPL. If you stop it at other times, updates that are made to the local node might not be sent to the other nodes.
- Ensure that all users or applications that update the RACF database are finished before issuing the STOP command.

Issuing options

The following table identifies the eligible options for issuing the STOP command:

<table>
<thead>
<tr>
<th>As a RACF TSO command?</th>
<th>As a RACF operator command?</th>
<th>With command direction?</th>
<th>With automatic command direction?</th>
<th>From the RACF parameter library?</th>
</tr>
</thead>
<tbody>
<tr>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
</tbody>
</table>

For information on issuing this command as a RACF operator command, see "Rules for entering RACF operator commands" on page 21. You must be logged on to the console to issue this command as a RACF operator command.

Related commands

- To restart the RACF subsystem address space, use the MVS START command.
- To restart a function in the RACF subsystem address space, see "RESTART (Restart RRSF functions)" on page 522. You must restart the RACF subsystem address space before restarting functions within the RACF subsystem address space.

Authorization required

When issuing the STOP command as a RACF operator command, you might require sufficient authority to the proper resource in the OPERCMDS class. See z/OS Security Server RACF Security Administrator's Guide for further information.

Syntax

The complete syntax of the command is:

```
subsystem-prefix STOP
```
STOP

For information on issuing this command as a RACF operator command, see "Rules for entering RACF operator commands" on page 21. You must be logged on to the console to issue this command as a RACF operator command.

Parameters

subsystem-prefix

Specifies that the RACF subsystem is the processing environment of the command. The subsystem prefix can be either the installation-defined prefix for RACF (1–8 characters) or, if no prefix has been defined, the RACF subsystem name followed by a blank. If the command prefix was registered with CPF, you can use the MVS command D OPDATA to display it or you can contact your RACF security administrator.

The subsystem prefix is a required keyword for RACF operator commands.

Examples

Table 96. STOP Example

<table>
<thead>
<tr>
<th>Example</th>
<th>Operation</th>
<th>Command</th>
<th>Known</th>
<th>Defaults</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example</td>
<td>User AMT02 wants to shut down the RACF subsystem address space in an orderly manner without losing any remote RRSF requests.</td>
<td>@STOP</td>
<td>The RACF subsystem prefix is @.</td>
<td>None.</td>
</tr>
</tbody>
</table>
TARGET (Define RRSF nodes)

Purpose

Use the TARGET command to:

- List the controls and operational characteristics of the specified target RRSF nodes.
- Specify the name of the target RRSF node.
- Request an operational state for connection to the target RRSF node.
- Delete an RRSF node from the set of target nodes known to the local node.
- Specify a description of the target RRSF node.
- Purge the workspace data sets managed by RRSF in the RACF subsystem address space.
- Specify the protocol type for the transport mechanism to be used in communication between the two RRSF remote nodes.
- Specify a prefix for the workspace data sets allocated by and used by RRSF for each target node.
- Specify the characteristics of the workspace data sets associated with the node being defined to RRSF.

Note: You might find it useful to fill out the “Configuration Worksheet” or read the chapter on RRSF in the z/OS Security Server RACF System Programmer’s Guide to help you determine the information you need to issue the TARGET command.

Issuing options

The following table identifies the eligible options for issuing the TARGET command:

<table>
<thead>
<tr>
<th>As a RACF TSO command?</th>
<th>As a RACF operator command?</th>
<th>With command direction?</th>
<th>With automatic command direction?</th>
<th>From the RACF parameter library?</th>
</tr>
</thead>
<tbody>
<tr>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>

For information on issuing this command as a RACF operator command, see "Rules for entering RACF operator commands" on page 21. You must be logged on to the console to issue this command as a RACF operator command.

Related commands

To set RRSF operational controls, see "SET" on page 567.

Authorization required

When issuing the TARGET command as a RACF operator command, you might require sufficient authority to the proper resource in the OPERCMDS class. See z/OS Security Server RACF Security Administrator’s Guide for further information.

Syntax

For the key to the symbols used in the command syntax diagrams, see "Syntax of RACF commands and operands" on page 9. The complete syntax of the TARGET command is:
### Parameters

**subsystem-prefix**

Specifies that the RACF subsystem is the processing environment of the command. The subsystem prefix can be either the installation-defined prefix for RACF (1–8 characters) or, if no prefix has been defined, the RACF subsystem name followed by a blank. If the command prefix was registered with CPF, you can use the MVS command D OPDATA to display it or you can contact your RACF security administrator.

Only specify the subsystem prefix when issuing this command as a RACF operator command. The subsystem prefix is required when issuing RACF operator commands.

**DELETE | DORMANT | OPERATIVE**

**DELETE**

Specifies that the target node is to be deleted from the set of known targets.

Subsequent attempts to perform operations requiring the existence of the deleted target fail and a message is issued.

When a node is deleted, any workspace data sets that are currently allocated are deallocated. If the workspace data sets are empty, they are also deleted. The local node can only be deleted after all other nodes are deleted.

The local system, the system represented by the target command whose SYSNAME matches the CVTSNAME of the system the command is to run on, cannot be deleted until all other TARGET definitions have been deleted.

The local MAIN system cannot be deleted until all of its remote target definitions have been deleted.
The MAIN system of a remote multisystem node cannot be deleted until all other peer systems of that remote multisystem node are deleted.

The DELETE keyword cannot be specified for a node whose connection is operative. The connection must first be made dormant.

**Notes:**
1. The DELETE keyword cannot be specified with the DORMANT or the OPERATIVE keywords. The only keywords that can be issued with the DELETE keyword are NODE, LOCAL, PURGE and SYSNAME.
2. If the PURGE keyword is specified with the DELETE keyword, the PURGE keyword is processed first regardless of the order in which the keywords were specified.

**DORMANT**
Specifies that the connection to the target is to be made inactive.
While a node is dormant, RRSF queues requests for the node and holds them on the OUTMSG queue until the connection to the node is made operative by an authorized operator issuing a TARGET OPERATIVE command.

**Notes:**
1. The DORMANT keyword cannot be specified with the DELETE or the OPERATIVE keyword.
2. If the PURGE keyword is specified with the DORMANT keyword, the DORMANT keyword is processed first regardless of the order in which the keywords were specified.
3. If either the remote node and/or the local node is a multisystem node, you must define a MAIN system to multisystem node before making the conversation dormant.

**OPERATIVE**
Specifies that the connection to the target node is to be made active.
As a result, all new and previously held requests for the node are sent.
The OPERATIVE keyword can only activate the connection to the target node if sufficient information has been provided on the WORKSPACE and PROTOCOL keywords and the PREFIX keyword has been specified.

**Notes:**
1. The OPERATIVE keyword cannot be specified with the DELETE or the DORMANT keyword.
2. If either the remote node and/or the local node is a multisystem node, you must define a MAIN system to the multisystem node before making the conversation operative.
3. If the PURGE keyword is specified with the OPERATIVE keyword, the PURGE keyword is processed first regardless of the order in which the keywords were specified.
4. If a connection is already operative and the OPERATIVE keyword is specified, the connection to that node is refreshed. If the node is the local node and the APPC server has been registered, the server is reregistered. The APPC transaction program (TP program) is restarted for local and remote nodes.

**DESCRIPTION('description')**
Specifies a comment used to describe this target node.
The following rules apply to description:

- The maximum length of the description is 32 characters.
- The description can consist of any characters, and can be entered with or without single quotation marks.
- If parentheses, commas, blanks, or semicolons are part of the description, the character string must be enclosed in single quotation marks.
- If a single quotation mark is part of the description, and the entire character string is enclosed in single quotation marks, two single quotation marks must be entered for each single quotation mark within the character string.
- If the first character of the description is a single quotation mark, then the string must be entered within single quotation marks and two single quotation marks must be entered for the first character.
- If the description is entered in lowercase, it is changed to uppercase.

The description is displayed in the TARGET LIST output.

LIST
Lists the current operational and protocol attributes of the target specified on the NODE keyword or all targets (*).

For a multisystem node, the LIST keyword can display information about the node and about the systems that make up the node.

The LIST keyword is the default on the TARGET command in the following situations:

- No other keyword is specified,
- Only the NODE(*) keyword is specified
- Only the NODE(nodename) keyword is specified,
- Only the NODE(nodename) and SYSNAME(sysname) keywords are specified,
- Only the NODE(nodename) and SYSNAME(*) keywords are specified, or
- Only the NODE(*) and SYSNAME(*) keywords are specified.

If the LIST keyword is specified in combination with any other TARGET command keyword, the information displayed provides the operational and protocol attributes as they exist after the processing of the other keywords.

If a request to list information specifies NODE(nodename), then a detailed list for that one target node is provided. If a request to list information specifies NODE(*), then a detailed list, sorted by target node name, is provided for all known nodes including systems on a multisystem node. If the NODE keyword is not specified, then a less detailed summary list is provided for all target nodes known to the node the command runs on, sorted by target node name.

The LIST keyword can display a summary or detailed list of information for an RRSF node. The type of information displayed, depends on the following combinations of NODE and SYSNAME:

**NODE(****)
Displays a detailed list of every node and system

**NODE(nodename) SYSNAME(****)
Displays a detailed list of every system in the multisystem node

**NODE(nodename)
For a multisystem node, displays a summary list of every system in the multisystem node
TARGET

**NODE**(nodename)
For a single-system node, displays a detailed list of the single-system node.

**NODE**(nodename) **SYSNAME**(systemname)
Displays a detailed list of the particular system.

**NODE**(+) **SYSNAME**(+)
Displays a detailed list of every node and system.

**LOCAL**
Identifies the node specified on the NODE keyword as the local node and all other keywords as local node specifications.

Only one local node can be defined. Once you have defined a node as the local node, you do not need to specify LOCAL on subsequent TARGET commands issued for that node or for systems being added to the local multisystem node. The local node must be defined before attempting to make any other nodes operative since the local SYSTEM name is needed to allocate the workspace data sets.

If you do not specify LOCAL on the TARGET command, the specified node is defined as a remote node.

**MAIN**
Identifies the system named on the SYSNAME keyword as the main system in a multisystem RRSF node. You must issue a TARGET command identifying the main system for a multisystem node on each system in the multisystem node, and on each RRSF node that communicates with the multisystem node. You must designate the same system as the main system on the local node and on all nodes that communicate with it. You must identify the main system of a multisystem node before you make any systems in the multisystem node operative or dormant.

**NODE**(nodenamel+)**
Specifies the name of the RRSF node that is being defined or is already defined and is being listed or modified by the specified keywords.

All nodes that communicate with the local node and the local node itself must be defined with a TARGET command.

The specified node name must:
- Be 1–8 characters in length
- Begin with either A1–Z, #, $, or @
- Use only A–Z, 0–9, # (X'7B'), $ (X'5B'), or @ (X'7C') as the second through eighth characters

The asterisk (+) can only be specified with the LIST operand. If other keywords are specified with NODE(+), the command fails with an error message.

**PREFIX**(qualifier ...)**
Specifies the high-level qualifiers that RRSF uses for the workspace data set names of the INMSG and OUTMSG queues for the RRSF node defined by the TARGET command. The WORKSPACE keyword specifies the attributes of the workspace data sets.

The maximum length of the prefix is 19 characters. Multiple qualifiers can be specified, separated by periods. For example, **PREFIX**(qualifier1.qualifier2...qualifier) specifies multiple qualifiers that serve as the highest level qualifiers of the workspace data set names.
RRSF uses the PREFIX to allocate workspace data sets for the node specified by the TARGET commands. If the workspace data sets are already allocated, the prefix cannot be changed and an error message is issued. When selecting a prefix, ensure that the workspace data sets are protected by a RACF DATASET profile, and that the user ID associated with the RACF subsystem address space has authority to create and access them. It is recommended that all systems of the same multisystem node (whether LOCAL or REMOTE) have the same prefix.

PROTOCOL

Protocol information is required for communication with remote nodes. Do not specify the PROTOCOL keyword when the local node is to run in local mode. If you specify the PROTOCOL keyword, APPC is required for the local node to go OPERATIVE and unnecessary processing occurs.

APPC

Specifies that APPC is used as the network transport mechanism with the specified operands.

LUNAME(luname)

LUNAME is a required operand of the APPC keyword for all nodes if a remote node is to be made operative through the TARGET command. Luname is the logical unit name which is associated with the target RRSF node being described. It can be a 1–8 character unqualified LU name or a 1–17 character qualified LU name in the format netid.luname, where netid and luname are each 1–8 characters.

No validity checking is performed for LU name. Further, if the same LU name is defined on multiple TARGET commands, the first occurrence is the one used. For example, if two TARGET commands are issued for two different RRSF node names, but with the same LU name, the first RRSF node is associated with LU name and an error message is issued indicating that a second usage of LU name has been detected.

With the multisystem node support, the LU name can be modified only if the node is in the initial or defined state.

The information needed for this operand is in the SYS1.PARMLIB member APPCPMxx of the target RRSF node.

When building the data set name for the workspace files, TARGET uses only the second qualifier of the LU name. If this qualifier is not unique within the group of DASD devices shared by the local system, you are required to use the WDSQUAL keyword to supply a unique qualifier for the workspace data set name. For remote nodes, TARGET uses WDSQUAL as the third qualifier of the data set name so that the data set name appears as follows:

prefix.local-luname.[remote-luname OR WDSQUAL-value].{INMSG or OUTMSG}

The output of the TARGET LIST command contains the qualified LU name if it is available.

TPNAME(profile-name)

profile-name specifies the APPC transaction program (TP) profile. The TP profile to be used at the target can be specified as a character string, 1–64 characters in length.

The default is IRRACF.

Once the profile-name is specified for a node, it can only be changed if the node is first made dormant.
TARGET

**MODENAME**(mode-name)

*modename* is the mode name which designates the network properties for the session to be allocated. The APPC mode (logmode) used can be specified as an alphanumeric string, 1–8 characters in length. If no mode name is specified, RACF supplies a MODENAME default. However, the TARGET LIST output shows the MODENAME as `<NOT SPECIFIED>`. See [z/OS Communications Server: SNA Programmer’s LU 6.2 Guide](https://www.ibm.com/support/knowledgecenter/S5M185_7.3.0/com.ibm.zos Theater/zos_comserver_sna/index.html) for further information.

Once the mode name is specified for a node, it can only be changed if the node is first made dormant.

The default is IRRMODE.

**PURGE (INMSG | OUTMSG)**

Specifies that all entries for the specified INMSG or OUTMSG workspace data set for the specified target node are to be purged.

INMSG and OUTMSG workspace data sets can only be purged if the node is first made dormant.

**Note:** If the PURGE keyword is specified with the DELETE or OPERATIVE keyword, the PURGE keyword is processed first regardless of the order in which the keywords were specified.

If the PURGE keyword is specified with the DORMANT keyword, the DORMANT keyword is processed first regardless of the order in which the keywords were specified.

**SYSNAME(sysname | *)**

Identifies which system on a multisystem node the command pertains to. If the SYSNAME keyword is specified, the NODE keyword must also be specified. The SYSNAME keyword is required on TARGET commands for multisystem nodes. If it is not specified, RACF assumes that the node is a single-system node. The SYSNAME keyword is not required if the LIST keyword is specified or defaulted.

The *sysname* value is 1–8 characters long, and can contain the characters A–Z, 0–9, ‑, #, @. It should not have a numeric as the first character, because it is used as a data set qualifier for the local workspace data sets. The *sysname* value must match the value in the CVTSNAME field of the system that the TARGET command describes. The CVTSNAME is the SYSNAME specified in the IEASYSxx member of SYS1.PARMLIB.

You can specify an asterisk (*) on the SYSNAME keyword to represent all systems currently defined for the node name specified by the NODE keyword. The asterisk (*) is only allowed to be specified with the NODE, DORMANT, OPERATIVE, DELETE, PURGE, and LIST keywords. When specified with DORMANT, OPERATIVE, DELETE, or PURGE, the requested action is attempted for all systems defined for that multisystem node. You can specify an asterisk (*) on the SYSNAME keyword when the LIST keyword is specified, causing a list to be generated for each system in the multisystem node specified on the NODE keyword. If NODE(*) is specified, SYSNAME must be specified as SYSNAME(*) or omitted.

The SYSNAME keyword allows you to use a common set of TARGET commands on all the systems in a multisystem node. When the TARGET command is for a local node, and the OPERATIVE or DORMANT keyword is specified, RACF compares the SYSNAME specified on the TARGET command with the CVTSNAME for the system the command is to run on. If they do not
match, RACF does not process the OPERATIVE or DORMANT keyword. The same is true for defined systems whose SYSNAMEs do not match the CVTSNAME when SYSNAME(=) is specified with OPERATIVE or DORMANT for a local node. In addition, because a conversation should not exist between the systems of a multisystem node, RACF issues an informational message and places it in the SYSLOG. This message might help diagnose why an expected conversation was not established.

**WDSQUAL**(qualified)

Specifies the middle qualifier that RRSF uses for the workspace data set names of the INMSG and OUTMSG queues for the local RRSF node defined by the TARGET command. When specified on the TARGET command for a remote node, it specifies a replacement qualifier for the remote LU name as the third qualifier of the workspace data set names for that node. The WORKSPACE keyword specifies the attributes of the workspace data sets.

The maximum length of the workspace data set name qualifier is 8 characters, and the first character must be an alphabetic character.

RRSF uses the qualifier to allocate workspace data sets for the node specified by the TARGET commands. If the workspace data sets are already allocated, the qualifier cannot be changed and an error message is issued. When selecting a qualifier, ensure that the workspace data sets are protected by a RACF DATASET profile, and that the user ID associated with the RACF subsystem address space has authority to create and access them.

The naming convention for the workspace data sets created on a node as a result of a TARGET command for a remote node is:

```
prefix.local-luname.remote-luname-or-wdsqual.ds-identity
```

Where:

- **prefix**
  - Is a value you specify with the PREFIX keyword on the TARGET command.

- **local-luname**
  - Is the LU name of the local node.

- **remote-luname-or-wdsqual**
  - Is the LU name of the remote node if a workspace data set name qualifier is not specified on the TARGET command. If a workspace data set name qualifier is specified on the TARGET command for a remote node, then it is used instead of the LU name for that node.

- **ds-identity**
  - Is either INMSG or OUTMSG.

The naming convention for the workspace data sets created on a node as a result of a TARGET LOCAL command is:

```
prefix.sysname-or-wdsqual.ds-identity
```

Where:

- **prefix**
  - Is a value you specify with the PREFIX keyword on the TARGET command.

- **sysname-or-wdsqual**
  - Is the system name if the workspace data set name qualifier is not specified on the TARGET command. The SYSNAME value must match the value in
the CVTSNAME field for the system it identifies. If a workspace data set name qualifier is specified on the TARGET command, it is used instead of the SYSNAME value.

ds-identity
   Is either INMSG or OUTMSG.

WORKSPACE

The WORKSPACE keyword, combined with its operands, specifies information to be used during the allocation of two work data sets. The data sets are used by RRSF to control information and messages associated with work being processed by the RACF subsystem address space for the specified node. There are two data sets for each target node. No workspace data sets are allocated for peer systems in the same multisystem node or for non-main members between two multisystem nodes. For multisystem nodes, it is recommended that all workspace data sets are allocated on shared resources so that all member systems of the multisystem node can access any workspace data set.

The two data sets are allocated by RRSF using the information provided on the WORKSPACE and PREFIX keywords to determine their fully qualified names, data set size and other data-management-related information. RRSF provides the low level qualifiers of the data set names as follows:

- **INMSG**
  The INMSG data set is used by RRSF as a queue to keep track of requests that are associated with inbound work or work that is to be delivered.

- **OUTMSG**
  The OUTMSG data set is used by RRSF as a queue to keep track of all directed requests.
  The request remains in the data set until confirmation is received that the specified target node has received the request, and is then removed.

The naming convention for the workspace data sets created on a node as a result of a TARGET LOCAL command is:

```
prefix.sysname-or-wdsqual.ds-identity
```

Where:

- **prefix**
  Is a value you specify with the PREFIX keyword on the TARGET command.

- **sysname-or-wdsqual**
  Is the system name if a workspace data set name qualifier is not specified on the TARGET command. The SYSNAME must match the value in the CVTSNAME field for the system it identifies. If a workspace data set name qualifier is specified on the TARGET command, then it is used instead of the SYSNAME.

- **ds-identity**
  Is either INMSG or OUTMSG.

The naming convention for the workspace data sets for remote connections is:

```
prefix.local-luname.remote-luname-or-wdsqual.ds-identity
```

Where:

- **prefix**
  Is a value you specify with the PREFIX keyword on the TARGET command.
TARGET

local-luname
Is the LU name of the local node.

remote-luname-or-wdsqual
Is the LU name of the remote node if a workspace data set name qualifier is not specified on the TARGET command. If a workspace data set name qualifier is specified on the TARGET command for a remote node, then it is used instead of the LU name for that node.

ds-identity
Is either INMSG or OUTMSG.

You can specify either an SMS or non-SMS WORKSPACE. The STORCLAS, DATACLAS, and MGMTCLAS keywords apply to an SMS WORKSPACE. The VOLUME keyword applies to a non-SMS WORKSPACE. The FILESIZE keyword applies to both.

You cannot specify the VOLUME keyword with any of the SMS keywords (STORCLAS, MGMTCLAS, or DATACLAS).

STORCLAS(class-name)
Specifies the SMS storage class name. This must be specified if SMS is being used for the workspace.

DATACLAS(class-name)
Specifies the SMS data class name. This keyword applies if SMS is being used for the workspace.

MGMTCLAS(class-name)
Specifies the SMS management class name. This keyword applies if SMS is being used for the workspace.

VOLUME(volume-serial)
Specifies the volume serial number of the volume that contains the RRSF workspace data sets. This is primarily a configuration value that describes where the next new workspace data set is allocated.

The volume-serial specified must be a valid volume on the system where the TARGET command is issued.

FILESIZE(nnnnnnnnnl500)
Specifies that space allocated for the workspace data sets should be sufficient to contain nnnnnnnnn entries for each of the data sets.

The range of values that can be specified is 1–2 147 483 647.

FILESIZE(500) is the initial value.

Note: The value supplied for FILESIZE is used to supply a RECORDS parameter when RACF invokes Access Methods Services DEFINE CLUSTER to create RRSF work data sets. Although RACF allows a maximum FILESIZE value of 2 147 483 647, the maximum value allowed for the RECORDS parameter by a particular release of Access Methods Services might be different.

Note: The VSAM workspace data sets can be preallocated using the sample JCL provided in SAMPLIB member IRRSRRSF.

If you have preallocated the VSAM workspace data sets using the JCL provided in the IRRSRRSF SAMPLIB, those values remain in effect. Specifying values on the operands of the WORKSPACE keyword does not override those values.
Values specified on the operands are displayed by the TARGET LIST NODE(nodename) command. Be aware that these values cannot be the actual values used for allocation.

Examples

Table 97. TARGET examples

Example 1

**Operation** User ADMIN wants a summary list of the current operational attributes for all nodes in the RRSF configuration.

**Known** The RACF subsystem prefix is @.

**Command** @TARGET

**Defaults** Since no keywords are specified, LIST is the default. Since NODE is not specified, the command defaults to a summary listing for all target nodes known to the node the command runs on.

**Output** See Figure 75.

Example 2

**Operation** User ADMIN at POKMVSA wants to define POKMVSA as the local node for an RRSF configuration and make it operative.

**Known**
- The RACF subsystem prefix is @.
- POKMVSA has DFP non-SMS running.
- The volume that contains the workspace data sets is DASD01.
- The high-level qualifier for the workspace data sets is SYS1.RACF.
- The APPC LUNAME is MF1AP001.
- APPC and VTAM have been installed and configured.
- Since the LIST keyword is specified in combination with other TARGET keywords, the information displayed provides the operational and protocol attributes as they exist after the processing of the other keywords.

**Command** @TARGET NODE(POKMVSA) LOCAL DESCRIPTION('POUGHKEEPSIE MVS SYSTEM A') PREFIX(SYS1.RACF) WORKSPACE(VOLUME(DASD01)) PROTOCOL(APPC(LUNAME(MF1AP001))) OPERATIVE LIST

**Defaults**
- The APPC TP name defaults to IRRRACF.
- The modename defaults to the APPC default for the system.
- FILESIZE defaults to 500.

**Output** See Figure 75.

Example 3

**Operation** User ADMIN wants to purge the OUTMSG workspace data set for node POKMVSB.

**Known** The RACF subsystem prefix is @. POKMVSB must be dormant to purge the workspace data sets. When the PURGE and DORMANT keywords are specified together, the DORMANT keyword is processed first.

**Command** @TARGET NODE(POKMVSB) PURGE(OUTMSG) DORMANT

**Defaults** None.

**Results** User ADMIN receives an informational message.

- @IRRM021I RACF SUBSYSTEM PURGE OF NODE POKMVSB OUTMSG FILE SYS1.RACF.POKMVSA.OUTMSG IS COMPLETE.
- @IRRM021I RACF SUBSYSTEM TARGET COMMAND HAS COMPLETED SUCCESSFULLY.

Example 4

**Operation** User ADMIN wants to delete node POKMVSC from the set of known target nodes.

**Known** The RACF subsystem prefix is @. POKMVSC is already dormant.

**Command** @TARGET NODE(POKMVSC) DELETE

**Defaults** None.
Each form of this TARGET command gets the same output:

- NODE1 <target
- NODE1 <target list

00- NODE1 IRRM009I (-) LOCAL RRSF NODE NODE1 IS IN THE OPERATIVE ACTIVE STATE.
- IRRM009I (-) REMOTE RRSF NODE NODE2 IS IN THE OPERATIVE ACTIVE STATE
- IRRM009I (-) REMOTE RRSF NODE NODE3 IS IN THE OPERATIVE PENDING CONNECTION STATE.
- IRRM009I (-) REMOTE RRSF NODE NODE4 IS IN THE OPERATIVE PENDING CONNECTION STATE.
- IRRM009I (-) REMOTE RRSF NODE RSFNODE4 IS IN THE OPERATIVE PENDING CONNECTION STATE.
- IRRM009I (-) REMOTE RRSF NODE RSFNODE4 IS IN THE OPERATIVE PENDING CONNECTION STATE.

Target command for the local node

<TARGET N(NODE1) LIST
IRRM010I (-) RSFJ SUBSYSTEM PROPERTIES OF LOCAL RRSF NODE NODE1:
STATE - OPERATIVE ACTIVE
DESCRIPTION - <NOT SPECIFIED
PROTOCOL - APPC
LU NAME - MF1AP001
TP PROFILE NAME - IRRRACF
MODENAME - <NOT SPECIFIED
TIME OF LAST TRANSMISSION TO - <NONE>
TIME OF LAST TRANSMISSION FROM - <NONE>
WORKSPACE FILE SPECIFICATION
PREFIX - "RSFJ.WORK"
WDSQUAL - <NOT SPECIFIED
FILESIZE - 500
VOLUME - TEMP01
FILE USAGE
"RSFJ.WORK.NODE1.INMSG"
- CONTAINS 0 RECORD(S)
- OCCUPIES 1 EXTENT(S)
"RSFJ.WORK.NODE1.UTMSG"
- CONTAINS 0 RECORD(S)
- OCCUPIES 1 EXTENT(S)

Figure 75. Output for TARGET Command (Part 1 of 2)
TARGET command for a remote node

<TARGET N(NODE2) LIST
or
<TARGET N(NODE2)

IRRM010I (->) RSFJ SUBSYSTEM PROPERTIES OF REMOTE RRSF NODE NODE2:
  STATE - OPERATIVE ACTIVE
  DESCRIPTION - <NOT SPECIFIED>
  PROTOCOL - APPC
    LU NAME - MF2AP001
    TP PROFILE NAME - IRRRACF
    MODENAME - <NOT SPECIFIED>
  TIME OF LAST TRANSMISSION TO - <NONE>
  TIME OF LAST TRANSMISSION FROM - <NONE>
WORKSPACE FILE SPECIFICATION
  PREFIX - "RSFJ.WORK"
  WDSQUAL - <NOT SPECIFIED>
  FILESIZE - 500
  VOLUME - TEMP01
  FILE USAGE
    "RSFJ.WORK.MF1AP001.MF2AP001.INMSG"
      - CONTAINS 0 RECORD(S)
      - OCCUPIES 1 EXTENT(S)
    "RSFJ.WORK.MF1AP001.MF2AP001.OUTMSG"
      - CONTAINS 0 RECORD(S)
      - OCCUPIES 1 EXTENT(S)

Figure 75. Output for TARGET Command (Part 2 of 2)
Appendix A. Naming considerations for resource profiles

Profile definitions

In RACF, resource profiles contain a description of a resource, including the authorized users and the access authority of each user. Resource profiles can be discrete, generic, or, additionally for the DATASET class, fully qualified generic. Regardless of whether a resource profile name is discrete or generic, the format of the name must follow certain rules. For profiles that protect general resources these rules are described by the entries in the class descriptor table and vary from class to class. For profiles that protect data sets these rules are the same as those used by TSO except that the high-level qualifier must be a valid RACF-defined user ID or group name. Also, because the first qualifier is an id, RACF expects a data set name to have a minimum of two qualifiers. For a description of the TSO/E data set naming conventions, see z/OS TSO/E User's Guide.

Discrete profiles

A discrete profile can protect a single resource that has unique security requirements. A discrete profile matches the name of the resource it protects and cannot exist independently of the resource. In the DATASET class, if you delete the resource, you delete the profile.

For example, a profile protecting a resource named SMITH.REXX.EXEC in class DATASET would protect the data set named SMITH.REXX.EXEC.

Generic profiles

A generic profile can protect several resources that have a similar naming structure and security requirements. Specify generic characters in the profile name if you want to protect more than one resource with the same security requirements.

One or more of the following generic characters are allowed:

- Percent sign (%)
- Single asterisk (*)
- Double asterisk (**)  
- Ampersand (&)

Notes:
1. The double asterisk (**) is not allowed with the DATASET class if enhanced generic naming (EGN) is inactive.
2. The ampersand (&) is only for general resource profile names and only if the RACFVARS class is active. Resource profiles can be created to protect resource names with unlike names. See z/OS Security Server RACF Security Administrator's Guide for more information.

For example, a profile protecting a resource named SMITH.* in class DATASET would protect all of SMITH's data sets that did not have a more specific profile defined (NOEGN is in effect).

Fully-qualified generic profiles (DATASET class only)

A fully-qualified generic profile matches exactly the name of the data set it protects.
One reason to choose a fully-qualified generic profile for data set protection is that the profile is not deleted if the data set is deleted. If the data set is deleted and then recreated, the protection is there without creating another profile. Another reason is to protect multiple copies with one profile.

Determining RACF protection

Although multiple generic profiles can match a general resource name, only the most specific profile actually protects it. For example, AB.CD+, AB.CD.*, and AB.*.CD all match the data set AB.CD, but AB.CD.* protects the data set.

The best way to determine which profile is protecting a given resource is to use one of the list commands.

To find out what profile is protecting a general resource, enter the RLIST command:

RLIST class-name resource-name

which looks for a discrete profile. If none is found, and generic profile checking is in effect for the class, the generic profile which protects the resource is displayed.

To find out what profile is protecting your data set, enter:

LISTDSD DA('data-set-name')

which looks for a discrete profile. If none is found, and generic profile checking is in effect for the DATASET class, enter:

LISTDSD DA('data-set-name') GENERIC

which looks for a generic profile.

The rest of the appendix discusses the rules governing:

- Profile names for data sets
- Profile names for general resources.

Profile names for data sets

Table 98 on page 648 shows the use of generic characters before RACF Release 1.9 and with RACF Release 1.9 or later. The use of the generic character % has not changed.

Note: Depending on whether EGN is active, the ending * has different meanings. These are explained in more detail later in this section.

<table>
<thead>
<tr>
<th>Release of RACF</th>
<th>Ending .** Allowed</th>
<th>Middle .** Allowed</th>
<th>Beginning * Allowed</th>
<th>Middle * Allowed</th>
<th>Ending * Allowed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Starting With RACF 1.9 EGN on</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Starting With RACF 1.9 EGN off</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Prior to RACF 1.9 EGN on</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Prior to RACF 1.9 EGN off</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>
For naming profiles in the DATASET class you can use discrete, generic, or fully qualified generic names.

**Discrete profiles**

These are the same as TSO data set names (see z/OS TSO/E Command Reference, except that the high-level qualifier (or the qualifier supplied by a command installation exit) must be a valid RACF-defined user ID or group name.

**Generic profile rules—enhanced generic naming inactive**

In the DATASET class, you can use generic characters as follows:

- Specify % to match any single character in a data set name
- Specify * as follows:
  - As a character at the end of a data set profile name (for example, ABC.DEF*) to match zero or more characters until the end of the name, zero or more qualifiers until the end of the data set name, or both
  - As a qualifier at the end of a profile name (for example, ABC.DEF.*) to match one or more qualifiers until the end of the data set name
  - As a qualifier in the middle of a profile name (for example, ABC.*.DEF) to match any one qualifier in a data set name
  - As a character at the end of a qualifier in the middle of a profile name (for example, ABC.DE*.F) to match zero or more characters until the end of the qualifier in a data set name.

**Note:** For profiles in the DATASET class, the high-level qualifier of the profile name must not be, nor can it contain, a generic character—for example, *.ABC, AB%.B, and AB*.AB are not allowed.

Tables are provided to show the variety of profiles that can be created using generics, using enhanced generic naming, and what happens to the profile protection if enhanced generic naming is turned off.

**Table 99** and **Table 100** provide examples of data set names using generic naming. Enhanced generic naming has not been turned on (SETROPTS NOEGN, the default, is in effect).

**Table 101** and **Table 102** provide examples of data set names with enhanced generic naming (SETR EGN is on).

**Table 103** and **Table 104** provide examples of data set names if enhanced generic naming is turned off after being turned on. It is not recommended that you turn EGN off after you have turned it on.

**Table 99. Generic naming for data sets with enhanced generic naming inactive—Asterisk at the end**

<table>
<thead>
<tr>
<th>Profile Name</th>
<th>AB .CD*</th>
<th>AB .CD .*</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resources protected by the profile</td>
<td>AB.CD</td>
<td>AB.CD.EF</td>
</tr>
<tr>
<td></td>
<td>AB.CDEF</td>
<td>AB.CD.EF</td>
</tr>
<tr>
<td></td>
<td>AB.CD.EF</td>
<td>AB.CD.XY</td>
</tr>
<tr>
<td></td>
<td>AB.CD.XY</td>
<td>AB.CD.EF.GH</td>
</tr>
<tr>
<td></td>
<td>AB.CD.EF</td>
<td>AB.CD.EF.GH</td>
</tr>
</tbody>
</table>
Naming considerations for resource profiles

Table 99. Generic naming for data sets with enhanced generic naming inactive—Asterisk at the end (continued)

<table>
<thead>
<tr>
<th>Profile Name</th>
<th>AB.CD*</th>
<th>AB.CD.*</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resources not protected by the profile</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ABC.DEF</td>
<td></td>
<td>AB.CD</td>
</tr>
<tr>
<td>ABC.XY.XY.DEF</td>
<td></td>
<td>AB.CDEF</td>
</tr>
<tr>
<td>AB.CDEF</td>
<td></td>
<td>AB.XY.XY.DEF</td>
</tr>
</tbody>
</table>

Table 100. Generic naming for data sets with enhanced generic naming inactive—Asterisk or percent sign in the middle

<table>
<thead>
<tr>
<th>Profile Name</th>
<th>ABC.%EF</th>
<th>AB.*.CD</th>
<th>AB.CD*.*.EF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resources protected by the profile</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ABC.DEF</td>
<td></td>
<td>AB.CD</td>
<td>AB.CDEF.EF</td>
</tr>
<tr>
<td>ABC.XEF</td>
<td></td>
<td>AB.CDE.EF</td>
<td></td>
</tr>
<tr>
<td>AB.CD.CD</td>
<td></td>
<td>AB.CDEF.EF</td>
<td></td>
</tr>
<tr>
<td>AB.CDEF</td>
<td></td>
<td>AB.CDE.EF</td>
<td></td>
</tr>
<tr>
<td>ABC.DEFGHI</td>
<td></td>
<td>AB.CD.EF</td>
<td></td>
</tr>
<tr>
<td>ABC.DEF.GHI</td>
<td></td>
<td>AB.CDEF</td>
<td></td>
</tr>
<tr>
<td>ABC.DDEF</td>
<td></td>
<td>AB.DEF</td>
<td></td>
</tr>
<tr>
<td>AB.XY.XY.DEF</td>
<td></td>
<td>AB.XY.XY.DEF</td>
<td></td>
</tr>
</tbody>
</table>

Generic profile rules—enhanced generic naming active

The **enhanced generic naming** option applies only to data sets and allows you to use double asterisk (**) in the DATASET class. It also changes the meaning of the single asterisk (*) at the end of a profile name.

Your RACF security administrator activates enhanced generic naming by issuing the SETROPTS command with the EGN operand. SETROPTS EGN makes the rules for data set and general resource profiles consistent with each other. Additionally, generic profiles can be more precise, and the generic profile names are more similar to other IBM products.

New installations should set EGN on immediately.

The following rules apply if you have enhanced generic naming in effect.

Specify * as follows:

- As a character at the end of a data set profile name to match zero or more characters until the end of the qualifier.
- As a qualifier at the end of a profile name to match one qualifier until the end of the data set name.
  The meaning of an ending asterisk depends on whether the installation is using generic profiles with or without EGN.

Specify ** as follows:

- As either a middle or end qualifier in a profile name to match zero or more qualifiers. Only one occurrence of a double asterisk is allowed in a profile name.
  For example, ABC.DE.* is allowed; ABC.DE** is not allowed; and A..**.B..** is not allowed.
  RACF does not allow you to specify any generic characters in the high-level qualifier of a data set name.
Table 101 and Table 102 show examples of generic profile names you can create when enhanced generic naming is active, and the resources protected and not protected by those profiles.

Table 101. Generic data set profile names created with enhanced generic naming active—Asterisk and double asterisk at the end

<table>
<thead>
<tr>
<th>Profile Name</th>
<th>AB.CD*</th>
<th>AB.CD.*</th>
<th>AB.CD.**</th>
<th>AB.CD.*,**</th>
<th>AB.CD.<em>,</em>**</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resources protected by the profile</td>
<td>AB.CD</td>
<td>AB.CD.EF</td>
<td>AB.CD.XY</td>
<td>AB.CD</td>
<td>ABC.DEF</td>
</tr>
<tr>
<td>Resources not protected by the profile</td>
<td>AB.CD.EF</td>
<td>AB.CD.EF.GH</td>
<td>AB.CD.XY</td>
<td>AB.CD.EF</td>
<td>ABC.DEF</td>
</tr>
</tbody>
</table>

Table 102. Generic data set profile names created with enhanced generic naming active—Asterisk, double asterisk, or percent sign in the middle

<table>
<thead>
<tr>
<th>Profile Name</th>
<th>ABC.%EF</th>
<th>AB.*.CD</th>
<th>AB,**.CD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resources protected by the profile</td>
<td>ABC.DEF</td>
<td>AB.CD.CD</td>
<td>AB.CD</td>
</tr>
<tr>
<td>Resources not protected by the profile</td>
<td>ABC.DEFGHI</td>
<td>AB.CD</td>
<td>ABC.DEF</td>
</tr>
</tbody>
</table>

Note: Although multiple generic profiles might match a data set name, only the most specific actually protects the data set. For example, AB.CD*, AB.CD.**, and AB,**.CD all match the data set AB.CD, but AB.CD** protects the data set.

In general, given two profiles that match a data set, you can find the more specific one by comparing the profile name from left to right. Where they differ, a nongeneric character is more specific than a generic character. In comparing generics, a % is more specific than an *, and an * is more specific than **. Another way to determine the most specific is with the SEARCH command, as there are some rare exceptions to the general rule. SEARCH always lists the profiles in the order of the most specific to the least specific.

Data set profiles created before enhanced generic naming is activated continue to provide the same RACF protection after this option is activated.

If you protect resources with generic profiles while enhanced generic naming is active and then deactivate this option, your resources can no longer be protected.

Table 103 on page 652 and Table 104 on page 652 show examples of generic profiles created with enhanced generic naming active and the protection after deactivation.
Naming considerations for resource profiles

Table 103. After deactivating EGN—Asterisk and percent sign in the middle

<table>
<thead>
<tr>
<th>Profile Name</th>
<th>How RACF displays the name after EGN is deactivated</th>
<th>Resources protected by the profile after EGN is deactivated</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABC.%EF ABC.*.DEF</td>
<td>ABC.%EF</td>
<td>Same as before</td>
</tr>
</tbody>
</table>

Table 104. After deactivating EGN—Asterisk and double asterisk at the end

<table>
<thead>
<tr>
<th>Profile Name</th>
<th>How RACF displays the name after EGN is deactivated</th>
<th>Resources protected by the profile after EGN is deactivated</th>
</tr>
</thead>
<tbody>
<tr>
<td>AB.CD* AB.CD.* AB.CD.** AB.CD*.** AB.CD.*.**</td>
<td>AB.CD* AB.CD.* AB.CD. AB.CD* AB.CD.*</td>
<td>None None None Same as before Same as before</td>
</tr>
</tbody>
</table>

Choosing between discrete and generic profiles

- Choose a **generic** profile for one of the following reasons:
  - To protect more than one data set with the same security requirements. The data sets protected by a generic profile must have some identical characters in their names. The profile name contains one or more generic characters (* or %).
  - If you have a single data set that might be deleted, then recreated, and you want the protection to remain the same, you can create a fully qualified generic profile.
- Choose a **discrete** profile for the following reason:
  - To protect one data set with unique security requirements. The name of a discrete profile matches the name of the data set it protects.
  - While you could also use a fully qualified generic, you should do so with care. Generic profiles can cause performance problems if they are not used to protect several data sets.

If a data set is protected by both a generic profile and a discrete profile, the discrete profile takes precedence and sets the level of protection for the data set.

If a data set is protected by more than one generic profile, the most specific profile takes precedence and sets the level of protection for the data set.

Notes:
1. All the members of a partitioned data set (PDS) are protected by one profile (the profile that protects the data set).
2. For a generic profile, unit and volume information is ignored because the data sets that are protected under the generic profile can be on many different volumes.
Profile support
A generic profile might already exist under which the data set is protected. However, that profile might not provide the exact protection you want for your data set. In this case, you can create a more specific generic profile or a discrete profile for the data set.

All the components of a VSAM data set are protected by one profile (the profile that protects the cluster name). You do not need to create profiles that protect the index and data components of a cluster.

The protection offered by a generic profile is different, depending on the level of data management support installed on your system. Generic profiles protect all data sets that they apply to, including existing data sets and data sets to be created in the future.

A generic profile controls the creation of data sets. When a user creates a new tape or DASD data set that is protected by an existing generic profile, that profile must give the user ALTER authority. If the new data set is a group data set, the user must have either ALTER authority in the profile or CREATE authority in the group.

Data sets that are not RACF-indicated but are protected by a generic profile are not protected if they are transferred (in any way) or available (such as through shared DASD) to another system unless that system has all of the following:

- RACF protection
- The appropriate predefined generic profiles.

For more information, see "z/OS Security Server RACF Security Administrator’s Guide".

Profile names for general resources
For the syntax required for profiles in the DIRECTRY and FILE classes, see RACF Command Language Reference for your VM system. Table 105 shows the availability of generic characters before and with RACF Release 1.9 or later. The usage of the generic character $ has not changed.

Note: The ending asterisk has different meanings and is explained further in the appropriate sections.

<table>
<thead>
<tr>
<th></th>
<th>Double Asterisk Allowed in Beginning, Middle, or End</th>
<th>Middle Asterisk Allowed</th>
<th>Beginning Asterisk Allowed</th>
<th>Ending Asterisk Allowed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Starting With RACF 1.9</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Prior to RACF 1.9</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>

For naming general resources, you can use discrete or generic profiles. As mentioned before, discrete profile names exactly match the general resource name.

Valid generic characters are a percent sign (%), asterisk (*), double asterisk (**), and ampersand (&):

- Specify a percent sign to match any single character in a resource profile name.
- Specify a double asterisk once in a profile name as follows:
Naming considerations for resource profiles

- As the entire profile name to match all resource names in a class.
- As either a beginning, middle, or ending qualifier (for example, **.ABC, ABC..DEF, or ABC.*) to match zero or more qualifiers in a resource name.

**Note:** ** is always available for general resources. The SETROPTS EGN setting is exclusively for data sets.

- Specify an asterisk as follows:
  - As a qualifier at the beginning of a profile name to match any one qualifier in a resource name.
  - As a character at the end of a profile name (for example, ABC.DEF+) to match zero or more characters until the end of the resource name, zero or more qualifiers until the end of the resource name, or both.
  - As a qualifier at the end of a profile name (for example, ABC.DEF.*) to match one or more qualifiers until the end of the resource name.
  - As a qualifier in the middle of a profile name (for example, ABC.*.DEF) to match any one qualifier in a resource name.
  - As a character at the end of a qualifier in the middle of a profile name (for example, ABC.DE+.FGH) to match zero or more characters until the end of the qualifier in a resource name.

- Specify an ampersand as follows:
  - In a profile name to indicate that RACF is to use a profile in the RACFVARS class to determine the actual values to use for that part of the profile name.

**Notes:**

1. If a class in the class descriptor table specifies a value for key qualifiers other than zero, generic profiles created in that class cannot contain generic characters in the specified number of qualifiers at the beginning of a profile name in that class. For example, the VM DIRECTRY class specifies KEYQUAL=2, so no generic characters are allowed in the first 2 qualifiers of DIRECTRY profile names.

2. If a class in the class descriptor table specifies a value for key qualifiers other than zero, all discrete and generic profiles in that class must have at least $n$n$n+1$ qualifiers in the profile name. The number of qualifiers is determined by counting the number of period characters in the profile and adding one; the first character is not examined. Any generic characters must be in the $n$n$n+1$ qualifier or beyond. Examples of valid profile names for KEYQUALIFIERS(2) are:
   
   A.B.C
   A.B.*
   A.B.C.D*

   The FILE and DIRECTRY classes have different rules. For the syntax required for profiles in the DIRECTRY and FILE classes, see the appropriate RACF Command Language Reference for your VM system.

See z/OS Security Server RACF Security Administrator's Guide for the unique naming conventions of specific classes and for a discussion of the RACFVARS class. See also the product documentation (such as PSF or CICS) for the naming conventions of specific classes.
Restricted Use of %* in General Resources

New profiles with an ending %* are not allowed, nor are profiles named %*. The RDEFINE command returns an error message.

Existing profiles with an ending %* are usable, but they should be deleted before creating any new profiles with a middle or beginning * or **. The RALTER and RDELETE commands accepts %* to enable you to make the changes.

Instead of using an ending %*, create new profiles ending with %.*, ** or * for similar function (change AB.C%* to AB.C%.*, ** or AB.C*).

If you have existing profiles named %*, you should create new profiles (suggested name **).

Note: When creating the new profiles, consider using the FROM operand for continued use of the same access list.

Table 106, Table 107, and Table 108 give examples of generic profile names for general resources.

Table 106. Generic naming for general resources—Percent sign, asterisk, or double asterisk at the beginning

<table>
<thead>
<tr>
<th>Profile name</th>
<th>%.*AB</th>
<th>*.AB</th>
<th>**.AB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resources protected by the profile</td>
<td>B.*AB</td>
<td>AB.*AB</td>
<td>AB **.AB</td>
</tr>
<tr>
<td></td>
<td>A.*AB</td>
<td>ABC.*AB</td>
<td>A.*AB</td>
</tr>
<tr>
<td>Resources not protected by the profile</td>
<td>AB.*AB</td>
<td>AB.C.*AB</td>
<td>AB.C.*AB</td>
</tr>
</tbody>
</table>

Table 107. Generic naming for general resources—Asterisk or double asterisk at the end

<table>
<thead>
<tr>
<th>Profile name</th>
<th>A.*C</th>
<th>A.*CD</th>
<th>A.*CD.*CD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resources protected by the profile</td>
<td>AB.C</td>
<td>AB.C.DEF</td>
<td>AB.C.DEF.CD</td>
</tr>
<tr>
<td></td>
<td>AB.C.DEF</td>
<td>AB.C.DEF</td>
<td>AB.C.DEF</td>
</tr>
<tr>
<td>Resources not protected by the profile</td>
<td>ABC.*DEF</td>
<td>ABC.*DEF</td>
<td>ABC.*DEF</td>
</tr>
<tr>
<td></td>
<td>ABC.*XY.XY.DEF</td>
<td>ABC.*XY.XY.DEF</td>
<td>ABC.*XY.XY.DEF</td>
</tr>
</tbody>
</table>

Table 108. Generic naming for general resources—Asterisk, double asterisk, or percent sign in the middle

<table>
<thead>
<tr>
<th>Profile name</th>
<th>A.*%EF</th>
<th>A.*CD</th>
<th>A.*CD.*CD</th>
<th>A.*CD.*CD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resources protected by the profile</td>
<td>ABC.*DEF</td>
<td>AB.CD.CD</td>
<td>AB.CD.CD</td>
<td>AB.CD</td>
</tr>
<tr>
<td></td>
<td>ABC.*XEF</td>
<td>AB.CD.CD</td>
<td>AB.CD.CD</td>
<td>AB.X.C</td>
</tr>
</tbody>
</table>

Appendix A. Naming considerations for resource profiles 655
Table 108. Generic naming for general resources—Asterisk, double asterisk, or percent sign in the middle (continued)

<table>
<thead>
<tr>
<th>Profile name</th>
<th>ABC.%EF</th>
<th>AB.*.CD</th>
<th>AB.CD*.CD</th>
<th>AB.**.CD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resources not protected by the profile</td>
<td>ABC.DEFGHI</td>
<td>AB.CD.EF</td>
<td>AB.CD.XY</td>
<td>AB.CD.EF</td>
</tr>
<tr>
<td></td>
<td>ABC.DEF.GHI</td>
<td>AB.CD.EF</td>
<td>AB.CD.XY</td>
<td>AB.CDEF</td>
</tr>
<tr>
<td></td>
<td>AB.CDE</td>
<td>ABC.X.CD.EF</td>
<td>ABC.DEF</td>
<td>ABC.XY.CD</td>
</tr>
<tr>
<td></td>
<td>AB.X.Y.CD</td>
<td>ABC.XY</td>
<td>ABC.XY.XY.CD</td>
<td></td>
</tr>
</tbody>
</table>

Although multiple generic profiles might match a general resource name, only the most specific actually protects the resource. For example, AB.CD*, AB.CD.**, and AB.**.CD all match the general resource AB.CD, but AB.CD** protects it.

In general, given two profiles that match a general resource, you can find the more specific one by comparing the profile name from left to right. Where they differ, a nongeneric character is more specific than a generic character. In comparing generics, a percent sign is more specific than an asterisk, and an asterisk is more specific than double asterisk. Another way to determine the most specific is with the SEARCH command, as there are some rare exceptions to the general rule. SEARCH always lists the profiles in the order of the most specific to the least specific.

Permitting profiles for GENERICOWNER classes

GENERICOWNER gives an installation the ability of restricting CLAUTH users from creating profiles in a class. In order to do this, a top-level ** profile is defined. This profile is owned by the system administrator and this profile blocks all non-SPECIAL users from creating profiles. A permitting profile must be defined for each CLAUTH user. Each profile defines the subset of resources in the class that the user is allowed to create.

When a CLAUTH user attempts to define a resource, a search is made for a less-specific (permitting) profile that covers the profile being defined. This less-specific profile is a profile that matches the more specific profile name, character for character, up to the ending * or ** or ending contiguous % characters in the less-specific name.

This definition might appear simple, but is not exactly what you might expect in comparison to the preceding section.

Table 109. Permitting profile names containing asterisks (*)

<table>
<thead>
<tr>
<th>Profile name</th>
<th>AA.*</th>
<th>AA.**</th>
<th>AA*</th>
<th>A.*.B.**</th>
</tr>
</thead>
<tbody>
<tr>
<td>covered</td>
<td>AA.BB</td>
<td>AA.B.C</td>
<td>AA.%</td>
<td>AA.%</td>
</tr>
<tr>
<td></td>
<td>AA.BB</td>
<td>AA.B.C</td>
<td>AA.%</td>
<td>AA.%</td>
</tr>
<tr>
<td></td>
<td>AA.BB</td>
<td>AA.B.C</td>
<td>AA.%</td>
<td>AA.%</td>
</tr>
<tr>
<td></td>
<td>AAC.BB</td>
<td>ABC.BB</td>
<td>%AA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ABC.BB</td>
<td>ABC.BB</td>
<td>%AA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ABC.BB</td>
<td>ABC.BB</td>
<td>%AA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ABC.BB</td>
<td>ABC.BB</td>
<td>%AA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ABC.BB</td>
<td>ABC.BB</td>
<td>%AA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ABC.BB</td>
<td>ABC.BB</td>
<td>%AA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ABC.BB</td>
<td>ABC.BB</td>
<td>%AA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ABC.BB</td>
<td>ABC.BB</td>
<td>%AA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ABC.BB</td>
<td>ABC.BB</td>
<td>%AA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ABC.BB</td>
<td>ABC.BB</td>
<td>%AA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ABC.BB</td>
<td>ABC.BB</td>
<td>%AA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ABC.BB</td>
<td>ABC.BB</td>
<td>%AA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ABC.BB</td>
<td>ABC.BB</td>
<td>%AA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ABC.BB</td>
<td>ABC.BB</td>
<td>%AA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ABC.BB</td>
<td>ABC.BB</td>
<td>%AA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ABC.BB</td>
<td>ABC.BB</td>
<td>%AA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ABC.BB</td>
<td>ABC.BB</td>
<td>%AA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ABC.BB</td>
<td>ABC.BB</td>
<td>%AA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ABC.BB</td>
<td>ABC.BB</td>
<td>%AA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ABC.BB</td>
<td>ABC.BB</td>
<td>%AA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ABC.BB</td>
<td>ABC.BB</td>
<td>%AA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ABC.BB</td>
<td>ABC.BB</td>
<td>%AA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ABC.BB</td>
<td>ABC.BB</td>
<td>%AA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ABC.BB</td>
<td>ABC.BB</td>
<td>%AA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ABC.BB</td>
<td>ABC.BB</td>
<td>%AA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ABC.BB</td>
<td>ABC.BB</td>
<td>%AA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ABC.BB</td>
<td>ABC.BB</td>
<td>%AA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ABC.BB</td>
<td>ABC.BB</td>
<td>%AA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ABC.BB</td>
<td>ABC.BB</td>
<td>%AA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ABC.BB</td>
<td>ABC.BB</td>
<td>%AA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ABC.BB</td>
<td>ABC.BB</td>
<td>%AA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ABC.BB</td>
<td>ABC.BB</td>
<td>%AA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ABC.BB</td>
<td>ABC.BB</td>
<td>%AA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ABC.BB</td>
<td>ABC.BB</td>
<td>%AA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ABC.BB</td>
<td>ABC.BB</td>
<td>%AA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ABC.BB</td>
<td>ABC.BB</td>
<td>%AA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ABC.BB</td>
<td>ABC.BB</td>
<td>%AA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ABC.BB</td>
<td>ABC.BB</td>
<td>%AA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ABC.BB</td>
<td>ABC.BB</td>
<td>%AA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ABC.BB</td>
<td>ABC.BB</td>
<td>%AA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ABC.BB</td>
<td>ABC.BB</td>
<td>%AA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ABC.BB</td>
<td>ABC.BB</td>
<td>%AA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ABC.BB</td>
<td>ABC.BB</td>
<td>%AA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ABC.BB</td>
<td>ABC.BB</td>
<td>%AA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ABC.BB</td>
<td>ABC.BB</td>
<td>%AA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ABC.BB</td>
<td>ABC.BB</td>
<td>%AA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ABC.BB</td>
<td>ABC.BB</td>
<td>%AA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ABC.BB</td>
<td>ABC.BB</td>
<td>%AA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ABC.BB</td>
<td>ABC.BB</td>
<td>%AA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ABC.BB</td>
<td>ABC.BB</td>
<td>%AA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ABC.BB</td>
<td>ABC.BB</td>
<td>%AA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ABC.BB</td>
<td>ABC.BB</td>
<td>%AA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ABC.BB</td>
<td>ABC.BB</td>
<td>%AA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ABC.BB</td>
<td>ABC.BB</td>
<td>%AA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ABC.BB</td>
<td>ABC.BB</td>
<td>%AA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ABC.BB</td>
<td>ABC.BB</td>
<td>%AA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ABC.BB</td>
<td>ABC.BB</td>
<td>%AA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ABC.BB</td>
<td>ABC.BB</td>
<td>%AA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ABC.BB</td>
<td>ABC.BB</td>
<td>%AA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ABC.BB</td>
<td>ABC.BB</td>
<td>%AA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ABC.BB</td>
<td>ABC.BB</td>
<td>%AA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ABC.BB</td>
<td>ABC.BB</td>
<td>%AA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ABC.BB</td>
<td>ABC.BB</td>
<td>%AA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ABC.BB</td>
<td>ABC.BB</td>
<td>%AA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ABC.BB</td>
<td>ABC.BB</td>
<td>%AA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ABC.BB</td>
<td>ABC.BB</td>
<td>%AA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ABC.BB</td>
<td>ABC.BB</td>
<td>%AA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ABC.BB</td>
<td>ABC.BB</td>
<td>%AA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ABC.BB</td>
<td>ABC.BB</td>
<td>%AA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ABC.BB</td>
<td>ABC.BB</td>
<td>%AA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ABC.BB</td>
<td>ABC.BB</td>
<td>%AA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ABC.BB</td>
<td>ABC.BB</td>
<td>%AA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ABC.BB</td>
<td>ABC.BB</td>
<td>%AA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ABC.BB</td>
<td>ABC.BB</td>
<td>%AA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ABC.BB</td>
<td>ABC.BB</td>
<td>%AA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ABC.BB</td>
<td>ABC.BB</td>
<td>%AA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ABC.BB</td>
<td>ABC.BB</td>
<td>%AA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ABC.BB</td>
<td>ABC.BB</td>
<td>%AA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ABC.BB</td>
<td>ABC.BB</td>
<td>%AA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ABC.BB</td>
<td>ABC.BB</td>
<td>%AA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ABC.BB</td>
<td>ABC.BB</td>
<td>%AA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ABC.BB</td>
<td>ABC.BB</td>
<td>%AA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ABC.BB</td>
<td>ABC.BB</td>
<td>%AA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ABC.BB</td>
<td>ABC.BB</td>
<td>%AA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ABC.BB</td>
<td>ABC.BB</td>
<td>%AA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ABC.BB</td>
<td>ABC.BB</td>
<td>%AA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ABC.BB</td>
<td>ABC.BB</td>
<td>%AA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ABC.BB</td>
<td>ABC.BB</td>
<td>%AA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ABC.BB</td>
<td>ABC.BB</td>
<td>%AA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ABC.BB</td>
<td>ABC.BB</td>
<td>%AA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ABC.BB</td>
<td>ABC.BB</td>
<td>%AA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ABC.BB</td>
<td>ABC.BB</td>
<td>%AA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ABC.BB</td>
<td>ABC.BB</td>
<td>%AA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ABC.BB</td>
<td>ABC.BB</td>
<td>%AA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ABC.BB</td>
<td>ABC.BB</td>
<td>%AA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ABC.BB</td>
<td>ABC.BB</td>
<td>%AA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ABC.BB</td>
<td>ABC.BB</td>
<td>%AA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ABC.BB</td>
<td>ABC.BB</td>
<td>%AA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ABC.BB</td>
<td>ABC.BB</td>
<td>%AA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ABC.BB</td>
<td>ABC.BB</td>
<td>%AA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ABC.BB</td>
<td>ABC.BB</td>
<td>%AA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ABC.BB</td>
<td>ABC.BB</td>
<td>%AA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ABC.BB</td>
<td>ABC.BB</td>
<td>%AA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ABC.BB</td>
<td>ABC.BB</td>
<td>%AA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ABC.BB</td>
<td>ABC.BB</td>
<td>%AA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ABC.BB</td>
<td>ABC.BB</td>
<td>%AA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ABC.BB</td>
<td>ABC.BB</td>
<td>%AA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ABC.BB</td>
<td>ABC.BB</td>
<td>%AA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ABC.BB</td>
<td>ABC.BB</td>
<td>%AA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ABC.BB</td>
<td>ABC.BB</td>
<td>%AA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ABC.BB</td>
<td>ABC.BB</td>
<td>%AA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ABC.BB</td>
<td>ABC.BB</td>
<td>%AA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ABC.BB</td>
<td>ABC.BB</td>
<td>%AA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ABC.BB</td>
<td>ABC.BB</td>
<td>%AA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ABC.BB</td>
<td>ABC.BB</td>
<td>%AA</td>
<td></td>
</tr>
</tbody>
</table>
Naming considerations for resource profiles

Table 110. Permitting profile names containing percent signs (%)

<table>
<thead>
<tr>
<th>Profile name</th>
<th>AA.%</th>
<th>AA.%%</th>
<th>AA%</th>
<th>A.%%B</th>
</tr>
</thead>
<tbody>
<tr>
<td>covered</td>
<td>AA.B</td>
<td>AA.BB</td>
<td>AAC</td>
<td>A.%%B</td>
</tr>
<tr>
<td>not covered</td>
<td>AA.**</td>
<td>A%.A</td>
<td>A.A</td>
<td>A.A.B.CC</td>
</tr>
</tbody>
</table>

Commands to administer VM shared file system profiles

In z/OS, you cannot use the twelve RACF commands that were added for VM to manipulate shared file system (SFS) FILE and DIRECTRY profiles. However, you can use the existing general resource commands (RDEFINE, RALTER, RDELETE, RLIST, PERMIT, and SEARCH), to manipulate FILE and DIRECTRY profiles.

The profile name that you enter on the general resource commands must be in RACF format. The RACF format name contains periods as separators between all qualifiers (no colon), and no punctuation following the last qualifier. For FILE profiles, the file name and file type are the last two qualifiers in the name.

The following table summarizes the commands for administering SFS FILE and DIRECTRY profiles.

Table 111. Commands to administer shared file system (SFS) profiles

<table>
<thead>
<tr>
<th>Function</th>
<th>Command to be used on z/OS</th>
<th>Command provided on z/VM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add a DIRECTRY profile</td>
<td>RDEFINE DIRECTRY</td>
<td>ADDDIR ADIR</td>
</tr>
<tr>
<td>Add a FILE profile</td>
<td>RDEFINE FILE</td>
<td>ADDFILE AF</td>
</tr>
<tr>
<td>Alter a DIRECTRY profile</td>
<td>RALTER DIRECTRY</td>
<td>ALTDIR</td>
</tr>
<tr>
<td>Alter a FILE profile</td>
<td>RALTER FILE</td>
<td>ALTFILE ALF</td>
</tr>
<tr>
<td>Delete a DIRECTRY profile</td>
<td>RDELETE DIRECTRY</td>
<td>DELDIR DDIR</td>
</tr>
<tr>
<td>Delete a FILE profile</td>
<td>RDELETE FILE</td>
<td>DELFILE DF</td>
</tr>
<tr>
<td>List a DIRECTRY profile</td>
<td>RLIST DIRECTRY</td>
<td>LDIRECT LDIR</td>
</tr>
<tr>
<td>List a FILE profile</td>
<td>RLIST FILE</td>
<td>LFILE LF</td>
</tr>
<tr>
<td>Change access list in a DIRECTRY</td>
<td>PERMIT profile-name CLASS(DIRECTRY)</td>
<td>PERMDIR PDIR</td>
</tr>
<tr>
<td>profile</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Change access list in a FILE</td>
<td>PERMIT profile-name CLASS(FILE)</td>
<td>PERMFFILE PF</td>
</tr>
<tr>
<td>profile</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Search for a DIRECTRY profile</td>
<td>SEARCH CLASS(DIRECTORY)</td>
<td>SRDIR</td>
</tr>
<tr>
<td>Search for a FILE profile</td>
<td>SEARCH CLASS(FILE)</td>
<td>SRFILE SRF</td>
</tr>
</tbody>
</table>
Naming considerations for resource profiles
Appendix B. Supplied RACF resource classes

This appendix describes the general resource classes you can find in the supplied class descriptor table (CDT) and contains the following sections:

- "Supplied resource classes for z/OS systems"
- "Supplied resource classes for z/VM systems" on page 667

See [z/OS Security Server RACF Macros and Interfaces](#) to find the details (such as POSIX values) associated with the CDT entry for each class.

### Supplied resource classes for z/OS systems

Table 112 lists the supplied CDT classes that can be used on z/OS systems. See restrictions at the end of the table.

#### Table 112. Resource classes for z/OS systems

<table>
<thead>
<tr>
<th>Class name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALCSAUTH</td>
<td>Supports the Airline Control System/MVS (ALCS/MVS) product.</td>
</tr>
<tr>
<td>APPCLU</td>
<td>Verifying the identity of partner logical units during VTAM session establishment.</td>
</tr>
<tr>
<td>APPCPORT</td>
<td>Controlling which user IDs can access the system from a given LU (APPC port of entry). Also, conditional access to resources for users entering the system from a given LU.</td>
</tr>
<tr>
<td>APPCSERV</td>
<td>Controlling whether a program being run by a user can act as a server for a specific APPC transaction program (TP).</td>
</tr>
<tr>
<td>APPCSI</td>
<td>Controlling access to APPC side information files.</td>
</tr>
<tr>
<td>APPCTP</td>
<td>Controlling the use of APPC transaction programs.</td>
</tr>
<tr>
<td>APPL</td>
<td>Controlling access to applications.</td>
</tr>
<tr>
<td>CACHECLS</td>
<td>Contains profiles used for saving and restoring cache contents from the RACF database.</td>
</tr>
<tr>
<td>CBIND</td>
<td>Controlling the client’s ability to bind to the server.</td>
</tr>
<tr>
<td>CDT</td>
<td>Contains profiles for installation-defined classes for the dynamic CDT.</td>
</tr>
<tr>
<td>CFIELD</td>
<td>Contains profiles that define the installation’s custom fields.</td>
</tr>
<tr>
<td>CONSOLE</td>
<td>Controlling access to MCS consoles. Also, conditional access to other resources for commands originating from an MCS console.</td>
</tr>
<tr>
<td>DASDVOL</td>
<td>DASD volumes.</td>
</tr>
<tr>
<td>DBNFORM</td>
<td>Reserved for future IBM use.</td>
</tr>
<tr>
<td>DEVICES</td>
<td>Used by MVS allocation to control who can allocate devices such as:</td>
</tr>
<tr>
<td></td>
<td>• Unit record devices (printers and punches) (allocated only by PSF, JES2, or JES3)</td>
</tr>
<tr>
<td></td>
<td>• Graphics devices (allocated only by VTAM)</td>
</tr>
<tr>
<td></td>
<td>• Teleprocessing (TP) or communications devices (allocated only by VTAM)</td>
</tr>
<tr>
<td>DIGTCERT</td>
<td>Contains digital certificates and information related to them.</td>
</tr>
<tr>
<td>DIGTCRIT</td>
<td>Specifies additional criteria for certificate name filters.</td>
</tr>
<tr>
<td>DIGTNMAP</td>
<td>Mapping class for certificate name filters.</td>
</tr>
<tr>
<td>DIGTRING</td>
<td>Contains a profile for each key ring and provides information about the digital certificates that are part of each key ring.</td>
</tr>
</tbody>
</table>
### CDT classes

**Table 112. Resource classes for z/OS systems (continued)**

<table>
<thead>
<tr>
<th>Class name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DIRAUTH</td>
<td>Setting logging options for RACROUTE REQUEST=DIRAUTH requests. Also, if the DIRAUTH class is active, security label authorization checking is done when a user receives a message sent through the TPUT macro or the TSO SEND, or LISTBC commands.</td>
</tr>
<tr>
<td>DLFCLASS</td>
<td>The data lookaside facility.</td>
</tr>
<tr>
<td>FACILITY</td>
<td>Miscellaneous uses. Profiles are defined in this class so resource managers (typically elements of z/OS or z/VM) can check a user’s access to the profiles when the user takes some action. Examples are the profiles used to control execution of RACDCERT command functions and the profiles used to control privileges in the z/OS UNIX environment. RACF does not document all of the resources used in the FACILITY class by other products. For information on the FACILITY class resources used by a specific product (other than RACF itself), see that product’s documentation.</td>
</tr>
<tr>
<td>FIELD</td>
<td>Fields in RACF profiles (field-level access checking).</td>
</tr>
<tr>
<td>GDASDVOL</td>
<td>Resource group class for DASDVOL class.</td>
</tr>
<tr>
<td>GLOBAL</td>
<td>Global access checking table entry.</td>
</tr>
<tr>
<td>GMBR</td>
<td>Member class for the GLOBAL class.</td>
</tr>
<tr>
<td>GSDSF</td>
<td>Resource group class for SDSF class.</td>
</tr>
<tr>
<td>GTERMINL</td>
<td>Resource group class for TERMINAL class.</td>
</tr>
<tr>
<td>GXFACILI</td>
<td>Grouping class for XFACILIT resources.</td>
</tr>
<tr>
<td>IBMOPC</td>
<td>Controlling access to OPC/ESA subsystems.</td>
</tr>
<tr>
<td>IDIDMAP</td>
<td>Contains distributed identity filters created with the RACMAP command.</td>
</tr>
<tr>
<td>JESINPUT</td>
<td>Conditional access support for commands or jobs entered into the system through a JES input device.</td>
</tr>
<tr>
<td>JESJOBS</td>
<td>Controlling the submission and cancellation of jobs by job name.</td>
</tr>
<tr>
<td>JESSPOOL</td>
<td>Controlling access to job data sets on the JES spool (that is, SYSIN and SYSOUT data sets).</td>
</tr>
<tr>
<td>KEYSMSTR</td>
<td>Contains profiles that hold keys to encrypt data stored in the RACF database, such as LDAP BIND passwords and DCE passwords.</td>
</tr>
<tr>
<td>LDAPBIND</td>
<td>Contains the LDAP server URL, bind distinguished name, and bind password.</td>
</tr>
<tr>
<td>LOGSTRM</td>
<td>Controls system logger resources, such as log streams and the coupling facility structures associated with log streams.</td>
</tr>
</tbody>
</table>
| NODES | Controlling the following on MVS systems:  
  - Whether jobs are allowed to enter the system from other nodes  
  - Whether jobs that enter the system from other nodes have to pass user identification and password verification checks |
| NODMBR | Member class for the NODES class. |
| OPERCMDS | Controlling who can issue operator commands (for example, JES and MVS, and operator commands). |
| PMBR | Member class for the PROGRAM class. |
| PROGRAM | Protects executable programs. |
| PROPCNTL | Controlling if user ID propagation can occur, and if so, for which user IDs (such as the CICS or IMS main task user ID), user ID propagation is not to occur. |
## CDT classes

### Table 112. Resource classes for z/OS systems (continued)

<table>
<thead>
<tr>
<th>Class name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSFMPL</td>
<td>Used by PSF to perform security functions for printing, such as separator page labeling, data page labeling, and enforcement of the user printable area.</td>
</tr>
<tr>
<td>PTKTDATA</td>
<td>PassTicket key class enables the security administrator to associate a RACF secured signon secret key with a particular mainframe application that uses RACF for user authentication. Examples of such applications are IMS, CICS, TSO, z/VM, APPC, and MVS batch.</td>
</tr>
</tbody>
</table>
| RACFEVNT   | Contains profiles that control the following events:  
- LDAP change log notification for changes to certain RACF profiles  
- New password and password phrase enveloping for a given user. |
| RACFHC     | Used by IBM Health Checker for z/OS. Contains profiles that list the resources to check for each installation-defined health check. |
| RACFVARS   | RACF variables. In this class, profile names, which start with & (ampersand), act as RACF variables that can be specified in profile names in other RACF general resource classes. |
| RACGLIST   | Class of profiles that hold the results of RACROUTE REQUEST=LIST,GLOBAL=YES or a SETROPTS RACLIST operation. |
| RACHCMBR   | Used by IBM Health Checker for z/OS. Member class for the RACHCMBR class. |
| RDATALIB   | Used to control use of the R_datalib callable service (IRRSDL00 or IRRSDL64). |
| RRSFDATA   | Used to control RACF remote sharing facility (RRSF) functions. |
| RVARSMBR   | Member class for the RACFVARS class. |
| SCDMBR     | Member class for the SECDATA class. |
| SDSF       | Controls the use of authorized commands in the System Display and Search Facility (SDSF). See also GSDSF class. |
| SECDATA    | Security classification of users and data (security levels and security categories). |
| SECLABEL   | If security labels are used, and, if so, their definitions. |
| SECLMBR    | Member class for the SECLABEL class. |
| SERVAUTH   | Contains profiles used by servers to check a client’s authorization to use the server or to use resources managed by the server. Also, can be used to provide conditional access to resources for users entering the system from a given server. |
| SERVER     | Controlling the server’s ability to register with the daemon. |
| SMESSAGE   | Controlling to which users a user can send messages (TSO only). |
| SOMDOBJS   | Controlling the client’s ability to invoke the method in the class. |
| STARTED    | Used in preference to the started procedures table to assign an identity during the processing of an MVS START command. |
| SURROGAT   | If surrogate submission is allowed, and if allowed, which user IDs can act as surrogates. |
| SYSMVIEW   | Controlling access by the SystemView for MVS Launch Window to SystemView for MVS applications. |
| TAPEVOL    | Tape volumes. |
| TEMPDSN    | Controlling who can access residual temporary data sets. |
| TERMINAL   | Terminals (TSO or z/VM). See also GTERMINL class. |
## CDT classes

### Table 112. Resource classes for z/OS systems (continued)

<table>
<thead>
<tr>
<th>Class name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VTAMAPPL</td>
<td>Controlling who can open ACBs from non-APF authorized programs.</td>
</tr>
<tr>
<td>WRITER</td>
<td>Controlling the use of JES writers.</td>
</tr>
<tr>
<td>XFACLIT</td>
<td>Miscellaneous uses. Profile names in this class can be longer than 39 characters in length. Profiles are defined in this class so that resource managers (typically elements of z/OS) can check a user's access to the resources when the users take some action.</td>
</tr>
</tbody>
</table>

### CICS classes

<table>
<thead>
<tr>
<th>Class name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACICSPCT</td>
<td>CICS program control table.</td>
</tr>
<tr>
<td>BCICSPCT</td>
<td>Resource group class for the ACICSPCT class.</td>
</tr>
<tr>
<td>CCICSCMD</td>
<td>Used to verify that a user is permitted to use CICS system programmer commands such as INQUIRE, SET, PERFORM, and COLLECT.</td>
</tr>
<tr>
<td>CPSMOBJ</td>
<td>Used by CICSPlex® System Manager, which provides a central point of control when running multiple CICS systems, to determine operational controls within a CICS complex.</td>
</tr>
<tr>
<td>CPSMXMP</td>
<td>Used by CICSPlex System Manager to identify exemptions from security controls within a CICS complex.</td>
</tr>
<tr>
<td>DCICSDCT</td>
<td>CICS destination control table.</td>
</tr>
<tr>
<td>ECICSDCT</td>
<td>Resource group class for the DCICSDCT class.</td>
</tr>
<tr>
<td>FCICSFCT</td>
<td>CICS file control table.</td>
</tr>
<tr>
<td>GCICSTRN</td>
<td>Resource group class for TCIICSTRN class.</td>
</tr>
<tr>
<td>GCPSMOBJ</td>
<td>Resource grouping class for CPSMOBJ.</td>
</tr>
<tr>
<td>HCICSFCT</td>
<td>Resource group class for the FCICSFCT class.</td>
</tr>
<tr>
<td>JCICSSJCT</td>
<td>CICS journal control table.</td>
</tr>
<tr>
<td>KCICSSJCT</td>
<td>Resource group class for the JCICSSJCT class.</td>
</tr>
<tr>
<td>MCICSPPT</td>
<td>CICS processing program table.</td>
</tr>
<tr>
<td>NCICSPPT</td>
<td>Resource group class for the MCICSPPT class.</td>
</tr>
<tr>
<td>PCICSPSB</td>
<td>CICS program specification blocks (PSBs).</td>
</tr>
<tr>
<td>QCICSPSB</td>
<td>Resource group class for the PCICSPSB class.</td>
</tr>
<tr>
<td>RCICSSRES</td>
<td>CICS document templates.</td>
</tr>
<tr>
<td>SCICSTST</td>
<td>CICS temporary storage table.</td>
</tr>
<tr>
<td>TCICSTRN</td>
<td>CICS transactions.</td>
</tr>
<tr>
<td>UCICSTST</td>
<td>Resource group class for SCICSTST class.</td>
</tr>
<tr>
<td>VCICSCMD</td>
<td>Resource group class for the CCICSCMD class.</td>
</tr>
<tr>
<td>WCICSSRES</td>
<td>Resource group class for the RCICSSRES class.</td>
</tr>
</tbody>
</table>

### DB2 classes

<table>
<thead>
<tr>
<th>Class name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSNADM</td>
<td>DB2 administrative authority class.</td>
</tr>
<tr>
<td>DSNR</td>
<td>Controls access to DB2 subsystems.</td>
</tr>
<tr>
<td>GDSNBPF</td>
<td>Grouping class for DB2 buffer pool privileges.</td>
</tr>
<tr>
<td>GDSNCL</td>
<td>Grouping class for DB2 collection privileges.</td>
</tr>
<tr>
<td>GDSNDB</td>
<td>Grouping class for DB2 database privileges.</td>
</tr>
<tr>
<td>GDSNR</td>
<td>Grouping class for Java™ archive files (JARs).</td>
</tr>
<tr>
<td>GDSNPK</td>
<td>Grouping class for DB2 package privileges.</td>
</tr>
<tr>
<td>Class name</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>-------------</td>
</tr>
<tr>
<td>GDSNPN</td>
<td>Grouping class for DB2 plan privileges.</td>
</tr>
<tr>
<td>GDSNSC</td>
<td>Grouping class for DB2 schemas privileges.</td>
</tr>
<tr>
<td>GDSNSG</td>
<td>Grouping class for DB2 storage group privileges.</td>
</tr>
<tr>
<td>GDSNSM</td>
<td>Grouping class for DB2 system privileges.</td>
</tr>
<tr>
<td>GDSNSP</td>
<td>Grouping class for DB2 stored procedure privileges.</td>
</tr>
<tr>
<td>GDSNSQ</td>
<td>Grouping class for DB2 sequences.</td>
</tr>
<tr>
<td>GDSNTB</td>
<td>Grouping class for DB2 table, index, or view privileges.</td>
</tr>
<tr>
<td>GDSNTS</td>
<td>Grouping class for DB2 tablespace privileges.</td>
</tr>
<tr>
<td>GDSNUF</td>
<td>Grouping class for DB2 user-defined function privileges.</td>
</tr>
<tr>
<td>GDSNUT</td>
<td>Grouping class for DB2 user-defined distinct type privileges.</td>
</tr>
<tr>
<td>MDSNBP</td>
<td>Member class for DB2 buffer pool privileges.</td>
</tr>
<tr>
<td>MDSNCL</td>
<td>Member class for DB2 collection privileges.</td>
</tr>
<tr>
<td>MDSNDB</td>
<td>Member class for DB2 database privileges.</td>
</tr>
<tr>
<td>MDSNJR</td>
<td>Member class for Java archive files (JARs).</td>
</tr>
<tr>
<td>MDSNPK</td>
<td>Member class for DB2 package privileges.</td>
</tr>
<tr>
<td>MDSNPN</td>
<td>Member class for DB2 plan privileges.</td>
</tr>
<tr>
<td>MDSNSC</td>
<td>Member class for DB2 schema privileges.</td>
</tr>
<tr>
<td>MDSNSG</td>
<td>Member class for DB2 storage group privileges.</td>
</tr>
<tr>
<td>MDSNSM</td>
<td>Member class for DB2 system privileges.</td>
</tr>
<tr>
<td>MDSNSP</td>
<td>Member class for DB2 stored procedure privileges.</td>
</tr>
<tr>
<td>MDSNSQ</td>
<td>Member class for DB2 sequences.</td>
</tr>
<tr>
<td>MDSNTB</td>
<td>Member class for DB2 table, index, or view privileges.</td>
</tr>
<tr>
<td>MDSNTS</td>
<td>Member class for DB2 tablespace privileges.</td>
</tr>
<tr>
<td>MDSNUF</td>
<td>Member class for DB2 user-defined function privileges.</td>
</tr>
<tr>
<td>MDSNUT</td>
<td>Member class for DB2 user-defined distinct type privileges.</td>
</tr>
</tbody>
</table>

### DCE class

- **DCEUUIDS**: Used to define the mapping between a user’s RACF user ID and the corresponding DCE principal UUID.

### Enterprise Identity Mapping (EIM) class

- **RAUDITX**: Controls auditing for Enterprise Identity Mapping (EIM).

### Enterprise Java Beans classes

- **EJBROLE**: Member class for Enterprise Java Beans authorization roles.
- **GEJBRoLE**: Grouping class for Enterprise Java Beans authorization roles.
- **JAVA**: Contains profiles that are used by Java for z/OS applications to perform authorization checking for Java for z/OS resources.

### IMS classes

- **AIMS**: Application group names (AGN).
- **CIMS**: Command.
- **DIMs**: Grouping class for command.
- **FIMS**: Field (in data segment).
### Table 112. Resource classes for z/OS systems (continued)

<table>
<thead>
<tr>
<th>Class name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GIMS</td>
<td>Grouping class for transaction.</td>
</tr>
<tr>
<td>HIMS</td>
<td>Grouping class for field.</td>
</tr>
<tr>
<td>IIMS</td>
<td>Program specification block (PSB).</td>
</tr>
<tr>
<td>JIMS</td>
<td>Grouping class for program specification block (PSB).</td>
</tr>
<tr>
<td>LIMS</td>
<td>Logical terminal (LTERM).</td>
</tr>
<tr>
<td>MIMS</td>
<td>Grouping class for logical terminal (LTERM).</td>
</tr>
<tr>
<td>OIMS</td>
<td>Other.</td>
</tr>
<tr>
<td>PIMS</td>
<td>Database.</td>
</tr>
<tr>
<td>QIMS</td>
<td>Grouping class for database.</td>
</tr>
<tr>
<td>RIMS</td>
<td>Open Transaction Manager Access (OTMA) transaction pipe (TPIPE).</td>
</tr>
<tr>
<td>SIMS</td>
<td>Segment (in database).</td>
</tr>
<tr>
<td>TIMS</td>
<td>Transaction (trancode).</td>
</tr>
<tr>
<td>UIMS</td>
<td>Grouping class for segment.</td>
</tr>
<tr>
<td>WIMS</td>
<td>Grouping class for other.</td>
</tr>
</tbody>
</table>

#### Integrated Cryptographic Service Facility (ICSF) classes

<table>
<thead>
<tr>
<th>Class name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CRYPTOZ</td>
<td>Controls access to PKCS #11 tokens.</td>
</tr>
<tr>
<td>CSFKEYS</td>
<td>Controls access to ICSF cryptographic keys.</td>
</tr>
<tr>
<td>CSFSERV</td>
<td>Controls access to ICSF cryptographic services.</td>
</tr>
<tr>
<td>GCSFKEYS</td>
<td>Resource group class for the CSFKEYS class.</td>
</tr>
<tr>
<td>GXCSFKEY</td>
<td>Resource group class for the XCSFKEY class.</td>
</tr>
<tr>
<td>XCSFKEY</td>
<td>Controls the exportation of ICSF cryptographic keys.</td>
</tr>
</tbody>
</table>

#### Infoprint Server class

<table>
<thead>
<tr>
<th>Class name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRINTSRV</td>
<td>Controls access to printer definitions for Infoprint Server.</td>
</tr>
</tbody>
</table>

#### Information/Management (Tivoli Service Desk) classes

<table>
<thead>
<tr>
<th>Class name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GINFOMAN</td>
<td>Grouping class for Information/Management (Tivoli Service Desk) resources.</td>
</tr>
<tr>
<td>INFOMAN</td>
<td>Member class for Information/Management (Tivoli Service Desk) resources.</td>
</tr>
</tbody>
</table>

#### LFS/ESA classes

<table>
<thead>
<tr>
<th>Class name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LFSCLASS</td>
<td>Controls access to file services provided by LFS/ESA.</td>
</tr>
</tbody>
</table>

#### License Manager class

<table>
<thead>
<tr>
<th>Class name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ILMADMIN</td>
<td>Controls access to the administrative functions of IBM License Manager.</td>
</tr>
</tbody>
</table>

#### Lotus Notes for z/OS and Novell Directory Services for OS/390 classes

<table>
<thead>
<tr>
<th>Class name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NDSLINK</td>
<td>Mapping class for Novell Directory Services for OS/390 user identities.</td>
</tr>
<tr>
<td>NOTE LINK</td>
<td>Mapping class for Lotus Notes for z/OS user identities.</td>
</tr>
</tbody>
</table>

#### MQSeries® classes

<table>
<thead>
<tr>
<th>Class name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GMQADMIN</td>
<td>Grouping class for MQSeries administrative options.</td>
</tr>
<tr>
<td>GMQCHAN</td>
<td>Reserved for MQSeries.</td>
</tr>
<tr>
<td>GMQNLIST</td>
<td>Grouping class for MQSeries namelists.</td>
</tr>
<tr>
<td>GMQPROC</td>
<td>Grouping class for MQSeries processes.</td>
</tr>
<tr>
<td>Class name</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>GMQQUEUE</td>
<td>Grouping class for MQSeries queues.</td>
</tr>
<tr>
<td>MQADMIN</td>
<td>Protects MQSeries administrative options.</td>
</tr>
<tr>
<td>MQCHAN</td>
<td>Reserved for MQSeries.</td>
</tr>
<tr>
<td>MQCMDS</td>
<td>Protects MQSeries commands.</td>
</tr>
<tr>
<td>MQCONN</td>
<td>Protects MQSeries connections.</td>
</tr>
<tr>
<td>MQNLIST</td>
<td>Protects MQSeries namelists.</td>
</tr>
<tr>
<td>MQPROC</td>
<td>Protects MQSeries processes.</td>
</tr>
<tr>
<td>MQQUEUE</td>
<td>Protects MQSeries queues.</td>
</tr>
<tr>
<td>NETCMDS</td>
<td>NetView classes controlling which NetView commands the NetView operator can issue.</td>
</tr>
<tr>
<td>NETSPAN</td>
<td>Controlling which NetView commands the NetView operator can issue against the resources in this span.</td>
</tr>
<tr>
<td>NVASAPDT</td>
<td>NetView/Access Services.</td>
</tr>
<tr>
<td>PTKTVAL</td>
<td>Used by NetView/Access Services Secured Single Signon to store information needed when generating a PassTicket.</td>
</tr>
<tr>
<td>RMTOPS</td>
<td>NetView Remote Operations.</td>
</tr>
<tr>
<td>RODMMGR</td>
<td>NetView Resource Object Data Manager (RODM).</td>
</tr>
<tr>
<td>KERBLINK</td>
<td>z/OS Network Authentication Service classes</td>
</tr>
<tr>
<td>REALM</td>
<td>Mapping class for user identities of local and foreign principals.</td>
</tr>
<tr>
<td>SUBSYSNM</td>
<td>Used to define the local and foreign realms.</td>
</tr>
<tr>
<td>MGMTCLAS</td>
<td>SMS management classes.</td>
</tr>
<tr>
<td>STORCLAS</td>
<td>SMS storage classes.</td>
</tr>
<tr>
<td>SUBSYSNM</td>
<td>Authorizes a subsystem (such as a particular instance of CICS) to open a VSAM ACB and use VSAM record level sharing (RLS) functions.</td>
</tr>
<tr>
<td>ROLE</td>
<td>Tivoli classes</td>
</tr>
<tr>
<td>TMEADMIN</td>
<td>Specifies the complete list of resources and associated access levels that are required to perform the particular job function this role represents and defines which RACF groups are associated with this role.</td>
</tr>
<tr>
<td>ACCTNUM</td>
<td>Maps the user IDs of Tivoli administrators to RACF user IDs.</td>
</tr>
<tr>
<td>PERFGRP</td>
<td>TSO account numbers.</td>
</tr>
<tr>
<td>TSOAUTH</td>
<td>TSO performance groups.</td>
</tr>
<tr>
<td>TSOPROC</td>
<td>TSO user authorities such as OPER and MOUNT.</td>
</tr>
<tr>
<td>TSOPROC</td>
<td>TSO logon procedures.</td>
</tr>
<tr>
<td>GMRXADMIN</td>
<td>WebSphere MQ classes</td>
</tr>
<tr>
<td>GMRXNLIST</td>
<td>Grouping class for WebSphere MQ administrative options.</td>
</tr>
<tr>
<td>GMRXNLIST</td>
<td>Grouping class for WebSphere MQ namelists.</td>
</tr>
<tr>
<td>GMRXPROC</td>
<td>Grouping class for WebSphere MQ processes.</td>
</tr>
<tr>
<td>GMRXQUEUE</td>
<td>Grouping class for WebSphere MQ queues.</td>
</tr>
<tr>
<td>GMRXTOPIC</td>
<td>Grouping class for WebSphere MQ topics.</td>
</tr>
</tbody>
</table>
### CDT classes

**Table 112. Resource classes for z/OS systems (continued)**

<table>
<thead>
<tr>
<th>Class name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MXADMIN</td>
<td>Protects WebSphere MQ administrative options.</td>
</tr>
<tr>
<td>MXNLIST</td>
<td>Protects WebSphere MQ namelists.</td>
</tr>
<tr>
<td>MXPROC</td>
<td>Protects WebSphere MQ processes.</td>
</tr>
<tr>
<td>MXQUEUE</td>
<td>Protects WebSphere MQ queues.</td>
</tr>
<tr>
<td>MXTOPIC</td>
<td>Protects WebSphere MQ topics.</td>
</tr>
</tbody>
</table>

**z/OS UNIX classes**

<table>
<thead>
<tr>
<th>Class name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DIRACC</td>
<td>Controls auditing (using SETROPTS LOGOPTIONS) for access checks for read/write access to z/OS UNIX directories. This class need not be active to control auditing.</td>
</tr>
<tr>
<td>DIRSRCH</td>
<td>Controls auditing (using SETROPTS LOGOPTIONS) of z/OS UNIX directory searches. This class need not be active to control auditing.</td>
</tr>
<tr>
<td>FSOBJ</td>
<td>Controls auditing (using SETROPTS LOGOPTIONS) for all access checks for z/OS UNIX file system objects except directory searches. Controls auditing (using SETROPTS AUDIT) of creation and deletion of z/OS UNIX file system objects. This class need not be active to control auditing.</td>
</tr>
<tr>
<td>FSSEC</td>
<td>Controls auditing (using SETROPTS LOGOPTIONS) for changes to the security data (FSP) for z/OS UNIX file system objects. This class need not be active to control auditing. When this class is active, it also controls whether ACLs are used during authorization checks to z/OS UNIX files and directories.</td>
</tr>
<tr>
<td>IPCOBJ</td>
<td>Controls auditing (using SETROPTS LOGOPTIONS) of access checks for interprocess communication (IPC) objects and changes to security information of IPC objects. Controls auditing (using SETROPTS AUDIT) of the creation and deletion of IPC objects. This class need not be active to control auditing.</td>
</tr>
<tr>
<td>PROCACT</td>
<td>Controls auditing (using SETROPTS LOGOPTIONS) of functions that look at data from, or affect the processing of, z/OS UNIX processes. This class need not be active to control auditing.</td>
</tr>
<tr>
<td>PROCESS</td>
<td>Controls auditing (using SETROPTS LOGOPTIONS) of changes to UIDs and GIDs of z/OS UNIX processes. Controls auditing (using SETROPTS AUDIT) of dubbing and undubbing of z/OS UNIX processes. This class need not be active to control auditing.</td>
</tr>
<tr>
<td>UNIXMAP</td>
<td>Contains profiles that are used to map z/OS UNIX UIDs to RACF user IDs and z/OS UNIX GIDs to RACF group names.</td>
</tr>
<tr>
<td>UNIXPRIV</td>
<td>Contains profiles that are used to grant z/OS UNIX privileges.</td>
</tr>
</tbody>
</table>

**Restrictions:**

1. Do not specify this class name on the GENCMD, GENERIC, and GLOBAL/NOGLOBAL operands of the SETROPTS command.
2. Do not specify this class name on the GLOBAL operand of SETROPTS or, if you do, the GLOBAL checking is not performed.
3. Do not specify this class name on the GENCMD and GENERIC operands of the SETROPTS command.
4. Do not specify this class name with any RACF command. This is a member class associated with a grouping class that has a special use.
5. Profiles are not allowed in this class.
Supplied resource classes for z/VM systems

Table 113 lists the supplied classes you can use on z/VM systems. These classes are primarily relevant if you share your RACF database with a z/VM system. See restrictions at the end of the table.

Table 113. Resource classes for z/VM systems

<table>
<thead>
<tr>
<th>Class name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DIRECTRY</td>
<td>Protection of shared file system (SFS) directories.</td>
</tr>
<tr>
<td>FACILITY</td>
<td>Miscellaneous uses. Profiles are defined in this class so resource managers (typically elements of z/OS or z/VM) can check a user's access to the profiles when the user takes some action. Examples are the profiles used to control execution of RACDCERT command functions and the profiles used to control privileges in the z/OS UNIX environment. RACF does not document all of the resources used in the FACILITY class by other products. For information on the FACILITY class resources used by a specific product (other than RACF itself), see that product's documentation.</td>
</tr>
<tr>
<td>FIELD</td>
<td>Fields in RACF profiles (field-level access checking).</td>
</tr>
<tr>
<td>FILE</td>
<td>Protection of shared file system (SFS) files.</td>
</tr>
<tr>
<td>GLOBAL</td>
<td>Global access checking.</td>
</tr>
<tr>
<td>GMBR</td>
<td>Member class for GLOBAL class.</td>
</tr>
<tr>
<td>GTERMINL</td>
<td>Terminals whose IDs do not fit into generic profile naming conventions.</td>
</tr>
<tr>
<td>PSFMPL</td>
<td>When class is active, PSF/VM performs separator and data page labeling as well as auditing.</td>
</tr>
<tr>
<td>PTKTDATA</td>
<td>PassTicket key class.</td>
</tr>
<tr>
<td>PTKTVAL</td>
<td>Used by NetView/Access Services Secured Single Signon to store information needed when generating a PassTicket.</td>
</tr>
<tr>
<td>RACFVARS</td>
<td>RACF variables. In this class, profile names, which start with &amp; (ampersand), act as RACF variables that can be specified in profile names in other RACF general resource classes.</td>
</tr>
<tr>
<td>RVARSMBR</td>
<td>Member class for RACFVARS.</td>
</tr>
<tr>
<td>SCDMBR</td>
<td>Member class for SECDATA class.</td>
</tr>
<tr>
<td>SECDATA</td>
<td>Security classification of users and data (security levels and security categories).</td>
</tr>
<tr>
<td>SECLABEL</td>
<td>If security labels are used and, if so, their definitions.</td>
</tr>
<tr>
<td>SFSCMD</td>
<td>Controls the use of shared file system (SFS) administrator and operator commands.</td>
</tr>
<tr>
<td>TAPEVOL</td>
<td>Tape volumes.</td>
</tr>
<tr>
<td>TERMINAL</td>
<td>Terminals (TSO or z/VM). See also GTERMINL class.</td>
</tr>
<tr>
<td>VMBATCH</td>
<td>Alternate user IDs.</td>
</tr>
<tr>
<td>VMBR</td>
<td>Member class for VMEVENT class.</td>
</tr>
<tr>
<td>VMCMD</td>
<td>Certain CP commands and other requests on z/VM.</td>
</tr>
<tr>
<td>VMEVENT</td>
<td>Auditing and controlling security-related events (called z/VM events) on z/VM systems.</td>
</tr>
<tr>
<td>VMLAN</td>
<td>Controls access to z/VM guest LANs and virtual switches.</td>
</tr>
<tr>
<td>VMMAC</td>
<td>Used in conjunction with the SECLABEL class to provide security label authorization for some z/VM events.</td>
</tr>
</tbody>
</table>
### CDT classes

Table 113. Resource classes for z/VM systems (continued)

<table>
<thead>
<tr>
<th>Class name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VMMDISK</td>
<td>z/VM minidisks.</td>
</tr>
<tr>
<td>VMNODE</td>
<td>RSCS nodes.</td>
</tr>
<tr>
<td>VMRDR</td>
<td>z/VM unit record devices (virtual reader, virtual printer, and virtual punch).</td>
</tr>
<tr>
<td>VMSEGMENT</td>
<td>Restricted segments, which can be named saved segments (NSS) and discontiguous saved segments (DCSS).</td>
</tr>
<tr>
<td>VXMBR</td>
<td>Member class for VMXEVENT class.</td>
</tr>
<tr>
<td>VMXEVENT</td>
<td>Auditing and controlling security-related events (called z/VM events) on z/VM systems.</td>
</tr>
<tr>
<td>VMPOSIX</td>
<td>Contains profiles used by OpenExtensions for z/VM.</td>
</tr>
<tr>
<td>WRITER</td>
<td>z/VM print devices.</td>
</tr>
</tbody>
</table>

Restrictions:

1. Do not specify this class name on the GENCMD, GENERIC, and GLOBAL/NOGLOBAL operands of the SETROPTS command.
2. Do not specify this class name on the GLOBAL operand of SETROPTS or, if you do, the GLOBAL checking is not performed.
3. Do not specify this class name with any RACF command. This is a member class associated with a grouping class that has a special use.
4. Profiles are not allowed in this class.
Appendix C. Accessibility

Accessibility features help a user who has a physical disability, such as restricted mobility or limited vision, to use software products successfully. The major accessibility features in z/OS enable users to:

- Use assistive technologies such as screen readers and screen magnifier software
- Operate specific or equivalent features using only the keyboard
- Customize display attributes such as color, contrast, and font size

Using assistive technologies

Assistive technology products, such as screen readers, function with the user interfaces found in z/OS. Consult the assistive technology documentation for specific information when using such products to access z/OS interfaces.

Keyboard navigation of the user interface

Users can access z/OS user interfaces using TSO/E or ISPF. Refer to z/OS TSO/E Primer, z/OS TSO/E User’s Guide, and z/OS ISPF User’s Guide Vol for information about accessing TSO/E and ISPF interfaces. These guides describe how to use TSO/E and ISPF, including the use of keyboard shortcuts or function keys (PF keys). Each guide includes the default settings for the PF keys and explains how to modify their functions.

z/OS information

z/OS information is accessible using screen readers with the BookServer/Library Server versions of z/OS books in the Internet library at: http://www.ibm.com/systems/z/os/zos/bkserv/
Notices

This information was developed for products and services offered in the USA.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
USA

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

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

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

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

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

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

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
Mail Station P300
2455 South Road
Poughkeepsie, NY 12601-5400
USA

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

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.

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

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrates 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.

This product contains code licensed from RSA Data Security Incorporated.

RSA
SECURE

672  z/OS V1R11.0 Security Server RACF Command Language Reference
Policy for unsupported hardware

Various z/OS elements, such as DFSMS™, HCD, JES2, JES3, and MVS, contain code that supports specific hardware servers or devices. In some cases, this device-related element support remains in the product even after the hardware devices pass their announced End of Service date. z/OS may continue to service element code; however, it will not provide service related to unsupported hardware devices. Software problems related to these devices will not be accepted for service, and current service activity will cease if a problem is determined to be associated with out-of-support devices. In such cases, fixes will not be issued.

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, other countries, or both.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

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

Microsoft® and Windows® are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

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

Numerics
64-bit Data Encryption Standard
SESSKEY operand 444, 502

A
access attempt
changing
for data set profile 99, 101
access authority
changing in resource profiles 267
data sets 13
access checking
field-level 467
list-of-groups 600
access list
conditional 263
copying from another profile 268
deleting from profile 269
deleting names from 267
displaying for data set profile 217
displaying for general resource profile 531
standard 263
ACCESS operand
PERMIT command 267
access to system
controlling
for existing user 178
for new user 86
restoring for user 169, 170, 190, 191
revoking for user 170, 171, 191, 192
terminals 455, 509
accessibility 669
account number for TSO
changing for user 175
for existing user 172, 173, 176, 177
for new user 83
ACCTNUM class
description 665
ACCTNUM suboperand
ADDCATEGORY operand 444, 502
ADD function
RACDCERT command 288
ADDCATEGORY operand
ADDS command 37
ADDUSER command 53
ALTDS command 97
ALTUSER command 127
ADDGROUP command authorization required 24
description 24
elements 31
syntax 24, 25
ADDGROUP suboperand
ALTUSER command 145
ADDMEM operand
RALTER command 409
RDEFINE command 467
ADDMSCAPE suboperand
ALTUSER command 160
ADDDOMAINS suboperand
ALTUSER command 129, 146
address lines
ADDS command
authorization required 33
description 33
elements 45
syntax 35
ADDTOKEN function
RACDCERT command 294
ADDUSER command
authorization required 49
description 48
elements 89
syntax 50
ADDVOL operand
ALTDS command 98
RALTER command 411
administration, RACF
classroom courses xiv
ADSP (automatic data set protection) attribute
activating or deactivating system-wide 588
adding to user profile 127
deleting from user profile 128
for new user 53
in user’s connect profile 188
ADSP operand
ADDS command 53
ALTUSER command 127
CONNECT command 188
RDEFINE command 467
SETROPTS command 588
AGE operand
SEARCH command 558
AIMS class
description 663
ALCSAUTH class
  description 659
alias data set name
  RACF restriction on using 16
ALL operand
  HELP command 212
  LISTDSD command 217
  RLST command 531
  SEARCH command 558
ALTDIR 1
ALTDSD command
  authorization required 95
  description 94
  examples 106
  syntax 95
ALTER function
  RACDCERT command 298
ALTFILE 1
ALTGROUP command
  authorization required 108
  description 108
  examples 118
  syntax 109
ALTGRP suboperand
  ADDGROUP command 72
  ALTUSER command 155
ALTMAP function
  RACDCERT command 301
ALTNM operand
  RACDCERT GENCERT command 338
ALTUSER command
  authorization required 121
  changing
    in user profile 137
  description 120
  examples 181
  operator information 155
  primary language 142
  secondary language 142
  syntax 123
ALTVOL operand
  ALTDSD command 98
  APPCLU class
  description 659
APPCPORT class
  description 659
APPSCERV class
  description 659
APPCSI class
  description 659
APPCTP class
  description 659
APPL class
  description 659
APPL operand
  DISPLAY command 207
  SIGNOFF command 629
APPLAUDIT operand
  SETROPTS command 588
APPLDATA operand
  RALTER command 412
APPLDATA operand (continued)
  RDEFINE command 474
  application data
    changing for general resource profile 412
    defining for general resource profile 474
    deleting
      from general resource profile 414
  application key
    encrypting 447, 504
    masking 447, 504
ASSIZEMAX operand
  ADDUSER command 66
ASSIZEMAX suboperand
  ALTUSER command 147
AT operand
  ADDGROUP command 26
  ADDSD command 38
  ADDUSER command 54
  ALTDSD command 99
  ALTGROUP command 111
  ALTUSER command 128
  CONNECT command 188
  DELDSD command 196
  DELGROUP command 201
  DELUSER command 204
  LISTDSD command 217
  LISTGRP command 230
  LISTUSER command 242
  PASSWORD command 259
PERMIT command 267
PHRASE command 259
RALTER command 414
RDEFINE command 476
RDELETE command 517
REMOVE command 521
RLST command 531
SEARCH command 559
SETROPTS command 589
attribute
  AUDITOR
    for existing user 128
    for new user 54
  CLAUTH (class authority)
    for existing user 131
    for new user 56
  EIM (Enterprise Identity Mapping)
    for existing user 137
    for new user 60
  group-AUDITOR
    in user's connect profile 188
  group-OPERATIONS
    in user's connect profile 189
    logging activities for 609
  group-SPECIAL
    in user's connect profile 192
    logging activities for 620
  GRPACC (group access)
    for existing user 138
    for new user 60
    in user's connect profile 189
attribute (continued)

OPERATIONS
for existing user 154
for new user 72
logging activities for 609

SPECIAL
for existing user 172
for new user 83
logging activities for 620

audit access level
adding to new data set profile 38
changing
for data set profile 99, 101
changing for general resource profile 414, 431
defining for general resource profile 476

AUDIT operand
ADDS command 38
ALTDS command 99
RALTER command 414
RDEFINE command 476
SETROPTS command 589

AUDITOR attribute
for existing user 128
for new user 54

AUDITOR operand
ADDS command 54
ALTDS command 128
CONNECT command 188

AUTH suboperand
ADDS command 73
ALTDS command 155
authority
required to issue RACF commands 1
summary 1

AUTHORITY operand
ADDS command 54
ALTDS command 128
CONNECT command 189

AUTHUSER operand
LISTDS command 218
RLIST command 531
AUTO request reception
delete from user profile 156

AUTO suboperand
ADDS command 73
ALTDS command 156

AUTOAPPL operand
SET command 569

AUTODIRECT operand
SET command 569

AUTOGID suboperand
ALTGROUP command 114
automatic data protection (ADSP) attribute
See ADSP (automatic data set protection) attribute
automatic direction of application updates 569
automatic TAPEVOL profile
altering 401
permitting access to 264

AUTOUID suboperand
ADDS command 66
ALTDS command 147

B
batch
submitting RACF TSO commands from 16
BCICSPCT class
description 662

BIND function
RACDCERT command 305

BINDDN suboperand
ADDS command 80
ALTDS command 167
RALTER command 441
RDEFINE command 499

BINDPW suboperand
ADDS command 81
ALTDS command 168
RALTER command 441
RDEFINE command 499
bypassing
recording of statistics 601

C
CACHECLS class
description 659
canceling
syntax rules for passwords 614
system-wide ADSP (automatic data set protection) attribute 588

CATDSNS operand
SETROPTS command 590

class
description 659

CFDEF operand
RALTER command 423
RDEFINE command 483
RLIST command 532

CFDEF segment
altering 415
defining 476

certification
RACDCERT GENCERT command 338

CFIELD class
description 659

certification
RACDCERT GENCERT command 338

CFIELD class
description 659

certification
RACDCERT GENCERT command 338

CFIELD class
description 659

certification
RACDCERT GENCERT command 338

CFIELD class
description 659

certification
RACDCERT GENCERT command 338

CFIELD class
description 659

certification
RACDCERT GENCERT command 338

CFIELD class
description 659

certification
RACDCERT GENCERT command 338
changing (continued)
password interval 259
password phrase interval 259
CHECKCERT function
RACDCERT command 308
choosing between discrete and generic profiles
using commands on MVS 652
CICS
general resource classes 662
CICS operand
ADDUSER command 54
ALTUSER command 128
LISTUSER command 242
CICS segment
user profile
defining 54
displaying 242
CIMS class
description 663
class descriptor table (CDT)
protection classes 1
supplied classes for z/OS systems 659
supplied classes for z/VM systems 667
class name
activating or deactivating general resource
class 591
changing general resource profile 407
deleting general resource profile 516
displaying general resource profile 530
specifying
for general resource profile 465
CLASS operand
PERMIT command 267
SEARCH command 560
CLASSACT operand
SETROPTS command 591
classes
activating 582
names of supplied general resource classes for
z/OS 659
names of supplied general resource classes for
z/VM 667
recording statistics 622
classroom courses, RACF xiv
CLAUTH (class authority) attribute
for existing user 131
for new user 56
CLAUTH operand
ADDUSER command 56
ALTUSER command 131
CLIST data set
creating 560
CLIST operand
SEARCH command 560
CMDSYS suboperand
ADDUSER command 73
ALTUSER command 156
CMVDVIOL operand
SETROPTS command 592
command response logging (continued)
for new user profile 74
COMMAND suboperand
ADDUSER command 84
ALTUSER command 173
command usage
logging for a user 177
commands
summary 1
COMPATMODE operand
SETROPTS command 592
conditional access list 263
CONNECT command
authorization required 186
description 186
examples 193
syntax 187
CONNECT function
RACDCERT command 312
connect group
ALTUSER command 138
CONNECT command 189
CONNECT group authority
description 13
connect profile
assigning group-related attributes 186
changing 120, 186
creating 48, 186
deleting 202
displaying with group profile 227
displaying with user profile 237
CONSNAME suboperand
ADDUSER command 64
ALTUSER command 144
CONSOLE class
description 659
console command system
for new user profile 73
for user profile 156
console message format
for new user profile 75
for user profile 158, 159
console message storage
for new user profile 76
for user profile 161
console operator command authority
for new user 73
console search key
for new user profile 74
for user profile 157
controlled program
defining 408, 466
-specifying access for 272
controlling
access to system for existing user 178
access to system for new user 86
access to system for terminal 454, 509
copying access lists 268
courses about RACF xiv
CPSMOBJ class
description 662
D

DASD data set
  displaying volume information for 564
  erase-on-scratch processing
    activating 593
    deactivating system-wide 594
  for existing data set 100
  for new data set 39
  searching a volume for 565

DASDVOL class
  description 659
  data application for DFP
    changing
      for group profile 112
      for user profile 136
    defining
      for new group profile 27
      for new user profile 59
  data application for LNOTES
    changing
      for user profile 142
  data application for NDS
    changing
      for user profile 143
  data class for DFP
    changing
      in group profile 112
      in user profile 136
    defining
      for new group profile 27
      for new user profile 59

DATA operand
  ADDGROUP command 27
  ADDSD command 38
  ADDUSER command 56
  ALTGROUP command 112
  ALTUSER command 136
  DATA operand
  changing
    for group profile 111
    for existing user profile 131
    for new group profile 26
    for new user profile 56
  displaying
    for group profile 230
    for user profile 242

CSDATA operand
  ADDGROUP command 26
  ADDUSER command 56
  ALTGROUP command 111
  ALTUSER command 131
  CSDATA operand
  defining
    for existing group profile 111
    for existing user profile 131
    for new group profile 26
    for new user profile 56
  displaying
    for group profile 230
    for user profile 242

CSDATA segment
  defining
    for existing group profile 111
    for existing user profile 131
    for new group profile 26
    for new user profile 56
  displaying
    for group profile 230
    for user profile 242

CSFKEYS class
  description 664

CSFSERV class
  description 664

CTL suboperand
  ADDUSER command 64
  ALTUSER command 144

current password
  changing 260

current password phrase
  changing 260

current RACF options
  displaying 603

custom fields
  changing CFDEF segment 423
  defining
    for existing group profile 111
    for existing user profile 131
    for new group profile 26
    for new user profile 56
  defining CFDEF segment 483
  deleting
    for existing group profile 111
    for existing user profile 132
  listing CFDEF profiles 532

Index 679
DATA Set profile

OVM profile

OVM for group data sets 29
permitting access to 263
searching for
all profiles 558
based on last reference 558
selected profiles 561
tape data set profile 41

DATAAPPL suboperand
ADDGROUP command 27
ADDUSER command 59
ALTGROUP command 112
ALTUSER command 136
database
deactivating or reactivating RACF 546

DATA CLAS suboperand
ADDGROUP command 27
ADDUSER command 59
ALTGROUP command 112
ALTUSER command 136
TARGET command 642
data set
DELETING the distinguished name used by the LDAP proxy server
ALTUSER command 167
RALTER command 441
deleting the password used by the LDAP proxy server
ALTUSER command 168
RALTER command 442
deleting the URL of the LDAP proxy server
ALTUSER command 167
RALTER command 441
deleting the distinguished name used by the LDAP proxy server

DELETE function

RACDCERT command 312
RALTER command 435

DELCATEGORY operand
ALTDS command 98
ALTUSER command 127
RALTER command 409

DELETE operand
PERMIT command 267
TARGET command 634
deleting
access lists from profile 269
console command system
from user profile 156
collection key
from user profile 157
data set profile 194
general resource profile 514
group profile 199
message level from user profile 158
names from access list 267
operator command authority from user profile 156
primary language 142
secondary language 142
security category
from data set profile 98
from general resource profile 409
security level
from data set profile 104
from general resource profile 443
from user profile 172
user profile 202
volume
from tape volume profile 411
data set profile (continued)

permits accessing to 263
searching for
all profiles 558
based on last reference 558
selected profiles 561
tape data set profile 41

DATAAPPL suboperand
ADDGROUP command 27
ADDUSER command 59
ALTGROUP command 112
ALTUSER command 136
database
deactivating or reactivating RACF 546

DATA CLAS suboperand
ADDGROUP command 27
ADDUSER command 59
ALTGROUP command 112
ALTUSER command 136
TARGET command 642
data set
DELETING the distinguished name used by the LDAP proxy server
ALTUSER command 167
RALTER command 441
deleting the password used by the LDAP proxy server
ALTUSER command 168
RALTER command 442
deleting the URL of the LDAP proxy server
ALTUSER command 167
RALTER command 441
deleting the distinguished name used by the LDAP proxy server

DELETE function

RACDCERT command 312
RALTER command 435

DELCATEGORY operand
ALTDS command 98
ALTUSER command 127
RALTER command 409

DELETE operand
PERMIT command 267
TARGET command 634
deleting
access lists from profile 269
console command system
from user profile 156
collection key
from user profile 157
data set profile 194
general resource profile 514
group profile 199
message level from user profile 158
names from access list 267
operator command authority from user profile 156
primary language 142
secondary language 142
security category
from data set profile 98
from general resource profile 409
security level
from data set profile 104
from general resource profile 443
from user profile 172
user profile 202
volume
from tape volume profile 411
data set profile (continued)

permits accessing to 263
searching for
all profiles 558
based on last reference 558
selected profiles 561
tape data set profile 41

DATAAPPL suboperand
ADDGROUP command 27
ADDUSER command 59
ALTGROUP command 112
ALTUSER command 136
database
deactivating or reactivating RACF 546

DATA CLAS suboperand
ADDGROUP command 27
ADDUSER command 59
ALTGROUP command 112
ALTUSER command 136
TARGET command 642
data set
DELETING the distinguished name used by the LDAP proxy server
ALTUSER command 167
RALTER command 441
deleting the password used by the LDAP proxy server
ALTUSER command 168
RALTER command 442
deleting the URL of the LDAP proxy server
ALTUSER command 167
RALTER command 441
deleting the distinguished name used by the LDAP proxy server

DELETE function

RACDCERT command 312
RALTER command 435

DELCATEGORY operand
ALTDS command 98
ALTUSER command 127
RALTER command 409

DELETE operand
PERMIT command 267
TARGET command 634
deleting
access lists from profile 269
console command system
from user profile 156
collection key
from user profile 157
data set profile 194
general resource profile 514
group profile 199
message level from user profile 158
names from access list 267
operator command authority from user profile 156
primary language 142
secondary language 142
security category
from data set profile 98
from general resource profile 409
security level
from data set profile 104
from general resource profile 443
from user profile 172
user profile 202
volume
from tape volume profile 411
DELFIL E 1
DELGROU P command
  authorization required 199
description 200
  example 201
  syntax 200
DELMAP function
  RACDCERT command 318
  RACMAP command 396
DELMEM operand
  RALTER command 411
DELMSCOPE suboperand
  ALTUSER command 160
DELOPCLASS suboperand
  ALTUSER command 129, 146
DELRING function
  RACDCERT command 321
DELTOKEN function
  RACDCERT command 323
DELUSER command
  authorization required 203
  description 202
  example 205
  syntax 204
DELVOL operand
  ALTDS D command 98
  RALTER command 411
DESCRIPTION operand
  TARGET command 635
DEST suboperand
  ADDUSER command 84
  ALTUSER command 173
destination of SYSOUT data set
  for existing user 173
  for new user 84
DEVICES class
  description 659
DFLTGRP operand
  ADDUSER command 59
  ALTUSER command 135
DFP operand
  ADDGROUP command 27
  ADDSD command 39
  ADDUSER command 59
  ALTDS D command 100
  ALTGROUP command 112
  ALTUSER command 135
  LISTDS D command 219
  LISTGROUP command 230
  LISTUSER command 243
DFP segment
  changing
    in group profile 112
    in user profile 135
  defining
    for new group profile 27
    for new user profile 59
  displaying
    for group profile 230
    for user profile 243
  DFP-managed data set
    displaying owner 219
    specifying owner 39, 100
  DFSMSdfp
    general resource classes 665
  DIGTCERT class
    description 659
  DIGTCRIT class
    description 659
  DIGTNMAP class
    description 659
  DIGTRING class
    description 659
  DIMS class
    description 663
  DIREC T class
    description 666
  DIRAUTH class
    description 660
  DIRECT class
    description 660
  DIRSRCH class
    description 666
disability 669
discrete profile
  choosing between discrete and generic profiles 652
  data set
    defining 648, 649
    deleting 196
    displaying 220, 532
general resource
    defining 465, 653
    deleting 516
    displaying 532
    naming 647
    searching for 558
DISPLAY command
  authorization required 206
  description 206
  examples 208
  syntax 207
displaying
  current RACF options 603
  data set profile 213
  general resource profile 525
  group profile 227
  user profile 237
distributed identity filter
  using the RACMAP command 392
DLF object
  listing
    general resource profiles 532
    retain after use 489
    specifying which can be accessed 489
DLFCLASS class
  description 660
DLFDATA object
  retaining after use 428
DLFDATA operand
  RALTER command 428
  RDEFINE command 489
DLFDATA operand (continued)
   RLIST command 532
DLFDATA segment
   defining 489
DOCSIGN operand
   RACDCERT GENCERT command 338
DOM request reception
   deleting from user profile 157
   for new user profile 73
   for user profile 156
DOM suboperand
   ADDUSER command 73
   ALTUSER command 156
DOMAIN operand
   RACDCERT GENCERT command 338
DOMAINS suboperand
   ADDUSER command 65
   ALTUSER command 145
DORMANT operand
   TARGET command 635
DSA operand
   RACDCERT GENCERT command 335
DSN operand
   RACDCERT EXPORT command 325
DSNADM class
   description 662
DSNR class
   description 662
dynamic class descriptor table 415, 476, 532, 583

E
ECICSDCT class
   description 662
EGN operand
   SETROPTS command 592
EIM
   general resource class 663
EIM (Enterprise Identity Mapping)
   for existing user 137
   for new user 60
EIM domain
   altering resource options 429
   options 489
EIM operand
   ADDUSER command 60
   ALTUSER command 137
   LISTUSER command 243
   RALTER command 429
   RDEFINE command 489
EIM segment
   altering 429
   defining 489
   displaying
      for user profile 243
EJBROLE class
   description 663
EMAIL operand
   RACDCERT GENCERT command 338
ENCRYPT operand (continued)
   RALTER command 436
   RDEFINE command 495
ENCRYPT suboperand
   ALTUSER command 139
   enhanced generic naming
      activating or deactivating 592
      for data set profile 650
Enterprise Identity Mapping
   general resource class 663
Enterprise Java Beans
   general resource classes 663
   envelopes
      listing user information about 238
ERASE operand
   ADDSD command 39
   ALTSDS command 100
   SETROPTS command 593
erase-on-scratch processing
   activating 593
   deactivating 594
   for existing DASD data set 100
   for new DASD data set 39
event display information
   for new user profile 75
   for user profile 159
exit routine
   RACF commands that provide 12
EXPIRED operand
   ALTUSER command 137
EXPIRES operand
   SEARCH command 559
EXPORT function
   RACDCERT command 325
extension logic
   authorityKeyIdentifier extension 328
   basicConstraints extension 328
   issuerAltName extension 328
   keyUsage extension 328
   subjectAltName extension 328
   subjectKeyIdentifier extension 328

F
FACILITY class
   description 660, 667
FCICSFCT class
   description 662
FCLASS operand
   ADDSD command 39
   ALTSDS command 100
   SETROPTS command 593
FGeneric operand
   ADDSD command 39
   PERMIT command 268
   RDEFINE command 491
FILE class
   description 660, 667
   field-level access checking 467
file sequence number for tape data set 39
FILEPROCMAX suboperand
   ADDUSER command 68
   ALTUSER command 150
FILESEQ operand
   ADDSD command 39
FILESIZE suboperand
   TARGET command 642
FILTER operand
   SEARCH command 561
FIMS class
   description 663
FORCE operand
   RACDCERT DELTOKEN command 323
   RACDCERT ROLLOVER command 381
   RACDCERT UNBIND command 384
FORMAT operand
   RACDCERT EXPORT command 325
FROM operand
   ADDSD command 39
   PERMIT command 268
   RDEFINE command 491
FROMICSF operand
   RACDCERT GENCERT command 335
FSOBJ class
   description 666
FSROOT suboperand
   ADDUSER command 77
FSSEC class
   description 666
   fully-qualified generic profile
   naming 648
FUNCTION operand
   HELP command 212
FVOLUME operand
   ADDSD command 40
   PERMIT command 269
   RDEFINE command 492

G
GCICSTRN class
   description 662
GCPSMOBJ class
   description 662
GCSFKEYS class
   description 664
GDASDVOL class
   description 660
GDG (generation data group)
   activating model profile for 608
GDSNBP class
   description 662
GDSNCL class
   description 662
GDSNDB class
   description 662
GDSNJR class
   description 662
GDSNPK class
   description 662
GDSNPN class
   description 663
GDSNSC class
   description 663
GDSNSG class
   description 663
GDSNSM class
   description 663
GDSNSP class
   description 663
GDSNSQ class
   description 663
GDSNTB class
   description 663
GDSNTS class
   description 663
GDSNUF class
   description 663
GDSNUT class
   description 663
GEJBROLE class
   description 663
GENCERT function
   RACDCERT command 332
GENCMD operand
   SETROPTS command 594
general resource class 662
   activating 591
   auditing for 589
   deactivating 591
   generic profile checking 595
   generic profile command processing 594
   global access checking 471, 599
   product use of
      CICS 662
      DB2 662
      DCE 663
      DFSMSdtp 665
      EIM 663
      Enterprise Identity Mapping 663
      Enterprise Java Beans 663
      ICSF 664
      IMS 663
      Infoprint Server 664
      Information/Management 664
      LFS/ESA 664
      License Manager 664
      Lotus Notes for z/OS 664
      MQSeries 664
      NetView 665
      Novell Directory Services for OS/390 664
      SMS 665
      Tivoli 665
      Tivoli Service Desk 664
      TSO 665
      WebSphere MQ 665
      z/OS Integrated Security Services Network
      Authentication Service 665
      z/OS UNIX 666
      recording statistics for 622
      supplied for z/OS 659

Index 683
<table>
<thead>
<tr>
<th>General Resource Class</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>General Resource Class (continued) supplied for z/VM 667</td>
<td>global access checking activating or deactivating 599</td>
</tr>
<tr>
<td>General Resource Profile</td>
<td>defining entry in table 470</td>
</tr>
<tr>
<td>changing 401</td>
<td>refreshing in-storage table 620</td>
</tr>
<tr>
<td>defining 460, 653</td>
<td>GLOBAL class description 660, 667</td>
</tr>
<tr>
<td>deleting 514</td>
<td>GLOBAL operand SETROPTS command 599</td>
</tr>
<tr>
<td>determining RACF protection 648</td>
<td>GLOBALAUDIT operand ALTDS command 101</td>
</tr>
<tr>
<td>displaying 525</td>
<td>RALTER command 431</td>
</tr>
<tr>
<td>permitting access to 263</td>
<td>GMBR class description 660, 667</td>
</tr>
<tr>
<td>searching for based on last reference 558</td>
<td>GMOADMIN class description 664</td>
</tr>
<tr>
<td>searching RACF database for 561</td>
<td>GMQLIST class description 664</td>
</tr>
<tr>
<td>using existing profile as model for 491</td>
<td>GMQPROC class description 664</td>
</tr>
<tr>
<td>Generic Character</td>
<td>GMQQUEUE class description 665</td>
</tr>
<tr>
<td>defining generic profile name 647</td>
<td>GMXADMIN description 665</td>
</tr>
<tr>
<td>GENERIC operand ADDSD command 40</td>
<td>GMXNLIST description 665</td>
</tr>
<tr>
<td>GENERIC operand in place of 40</td>
<td>GMXPROC description 665</td>
</tr>
<tr>
<td>GENERIC operand ADDSD command 40</td>
<td>GMXQUEUE description 665</td>
</tr>
<tr>
<td>ALTDSD command 101</td>
<td>GMXTOPIC description 665</td>
</tr>
<tr>
<td>DELDSD command 197</td>
<td>Group default for existing user 135</td>
</tr>
<tr>
<td>LISTDSD command 220</td>
<td>default for new user 59</td>
</tr>
<tr>
<td>PERMIT command 269</td>
<td>group-related user attributes</td>
</tr>
<tr>
<td>RLIST command 532</td>
<td>assigning for user 186</td>
</tr>
<tr>
<td>SEARCH command 558</td>
<td>maximum number of users in 188</td>
</tr>
<tr>
<td>SETROPTS command 595</td>
<td>Group Authority description 13</td>
</tr>
<tr>
<td>Generic Profile choosing between discrete and generic profiles 652</td>
<td>for existing user 128</td>
</tr>
<tr>
<td>Data Set defining, enhanced generic naming active 650</td>
<td>for new user 54</td>
</tr>
<tr>
<td>defining, enhanced generic naming inactive 649</td>
<td>in user’s connect profile 189</td>
</tr>
<tr>
<td>deleting 197</td>
<td>Group Data Set model profile processing 608</td>
</tr>
<tr>
<td>displaying 220</td>
<td>Group Name as new owner</td>
</tr>
<tr>
<td>using existing profile as generic 101</td>
<td>of data set profiles of removed user 521</td>
</tr>
<tr>
<td>using new profile as generic 40</td>
<td>as owner of connect profile 190</td>
</tr>
<tr>
<td>Displaying for a Data Set 213</td>
<td>of data set profile 102</td>
</tr>
<tr>
<td>General Resource defining 465, 653</td>
<td>of general resource profile 440</td>
</tr>
<tr>
<td>deleting 516</td>
<td>of group profile 117</td>
</tr>
<tr>
<td>displaying 532</td>
<td>of new data set profile 42</td>
</tr>
<tr>
<td>naming 647</td>
<td>of new general resource profile 498</td>
</tr>
<tr>
<td>Refreshing In-storage Profiles 620</td>
<td>of new group profile 30</td>
</tr>
<tr>
<td>searching for 558</td>
<td>of new user profile 78</td>
</tr>
<tr>
<td>Generic Profile Checking 597, 598</td>
<td>of user profile 164</td>
</tr>
<tr>
<td>activating or deactivating 595</td>
<td>Changing access to resource for 269</td>
</tr>
<tr>
<td>Generic Profile Command Processing activating or deactivating 594</td>
<td>Deleting group profile 200</td>
</tr>
<tr>
<td>GENERICOWNER operand SETROPTS command 597</td>
<td>displaying data set profiles for 218</td>
</tr>
<tr>
<td>GENLIST operand</td>
<td>SETROPTS command 597</td>
</tr>
<tr>
<td>SETROPTS command 598</td>
<td>GENREQ function RACDCERT command 341</td>
</tr>
<tr>
<td>GENREQ Function</td>
<td>GID operand SEARCH command 562</td>
</tr>
<tr>
<td>GID Operand</td>
<td>GID Suboperand ADDGROUP command 29</td>
</tr>
<tr>
<td>SEARCH command 562</td>
<td>ALTDGROUP command 28, 115, 116</td>
</tr>
<tr>
<td>GIMS Class</td>
<td>description 664</td>
</tr>
<tr>
<td>GINFOMAN Class</td>
<td>description 664</td>
</tr>
</tbody>
</table>
group name (continued)
displaying (continued)
group profile for 229
for existing group 111
for new group 26
for removing user from group 521
syntax 10
GROUP operand
ALTUSER command 138
CONNECT command 189
DISPLAY command 208
RALTER command
   =MEMBER operand 448
RDEFINE command
   =MEMBER operand 505
REMOVE command 521
SIGNOFF command 629
group profile
changing 108
defining 24
deleting 199
displaying 227
searching for based on last reference 558
group-AUDITOR attribute
in user's connect profile 188
group-OPERATIONS attribute
in user's connect profile 189
logging activities for 609
group-SPECIAL attribute
in user's connect profile 189
logging activities for 620
GRPACC (group access) attribute
for existing user 138
for new user 60
in user's connect profile 189
GRPACC operand
ADDUSER command 60
ALTUSER command 138
CONNECT command 189
GRPLIST operand
SETROPTS command 600
GSDFS class
description 660
GTERMINL class
description 660, 667
GXCSFKEY class
description 664
GXFACIL1 class
description 660

H
HANDSHAKE operand
   RACDCERT GENCERT command 337
HC suboperand
   ADDUSER command 74
   ALTUSER command 157
HCICSFCT class
description 662
HELP command
description 211
HELP command (continued)
examples 212
syntax 211
hierarchical storage manager (HSM)
TVTOC operand
   RALTER command 453
HIGHTRUST operand
   RACDCERT ADD command 288
   RACDCERT ALTER command 298
HIMS class
description 664
HISTORY operand
   LISTDSDS command 221
   RLIST command 533
HISTORY suboperand
   PASSWORD operand 609
hold class for TSO
   for existing user 174
   for new user 84
HOLDCLASS suboperand
   ADDUSER command 84
   ALTUSER command 174
HOME suboperand
   ADDUSER command 69, 77
   ALTUSER command 150

I
IC suboperand
   ADDUSER command 65
   ALTUSER command 145
ICSF (Integrated Cryptographic Service Facility)
general resource classes 664
ICSF operand
   RACDCERT ADD command 289
   RACDCERT GENCERT command 335
   RACDCERT IMPORT command 345
   RACDCERT REKEY command 374
   RALTER command 432
   RDEFINE command 492
   RLST command 533
ICSF segment
   altering 432
   defining 492
ICTX operand
   RALTER command 434
   RDEFINE command 494
   RLST command 533
ICTX segment
   altering 434
   defining 494
ID operand
   LISTDSDS command 218
   PERMIT command 269
IDNFILTER operand
   RACDCERT MAP command 366
IIMS class
description 664
ILMADMIN class
description 664
IMPORT function

RACDCERT command 344
IMS (Information Management System)
  general resource classes 663
in-storage profile
  SETROPTS GENLIST processing for 598
  SETROPTS RACLIST processing for 616
INACTIVE operand
  RVARY command 549
  SETROPTS command 601
INCLUDE operand
  SET command 573
indirect or symbolic VOLSER 37
INFOMAN class
  description 664
Infoprint Server
  general resource class 664
Information/Management
  general resource classes 664
INITSTATS operand
  SETROPTS command 601
installation exit routine
  RACF commands that provide 12
installation-defined data
  changing
    in data set profile 99
    in general resource profile 428
    in user profile 132
  defining
    for data set profile 38
    for new user profile 56
  deleting
    from general resource profile 428
  displaying
    for general resource profile 526
    from data set profile 213
    from user profile 237
INTERVAL operand
  PASSWORD command
    PHRASE command 259
INTERVAL suboperand
  PASSWORD operand 610
INTIDS suboperand
  ADDUSER command 74
  ALTUSER command 157
IP operand
  RACDCERT GENCERT command 338
IPCOBJ class
  description 666
ISPF panels
  advantages 16
  authorization, additional 17
  compared to RACF TSO commands 16
  not affected by SETROPTS LANGUAGE setting 603
  sample for password rules 17

J
JAVA class
  description 663
JCICSJCT class
  description 662
JES (job entry subsystem)
  activating or deactivating options for 601
  SETROPTS command 601
JESINPUT class
  description 660
JESJOBS class
  description 660
JESSPOOL class
  description 660
JIMS class
  description 664
job class for TSO
  for existing user 174
  for new user 84
job entry subsystem (JES)
  See JES (job entry subsystem)
JOBCLASS suboperand
  ADDUSER command 84
  ALTUSER command 174
JOBNAMES suboperand
  DLFDATA operand 489
JOIN group authority
  description 13

K
KCICSJCT class
  description 662
KERB operand
  ADDUSER command 60
  ALTUSER command 138
  LISTUSER command 243
  RALTER command 435
  RDEFINE command 495
  RLIST command 534
KERB segment
  display for user profile 243
KERBLINK class
  description 665
KERBLVL operand
  SETROPTS command 602
KERBNAME operand
  ADDUSER command 61
  RALTER command 436
  RDEFINE command 495
  ALTUSER command 139
  key
    for user profile 243
  See application key
KEY suboperand
  ADDUSER command 74
  ALTUSER command 157
  keyboard 669
logging
access attempt
  for existing general resource profile  414
  for new general resource profile  476
access attempts
  based on security level  622
  for existing data set profile  99
  for new data set profile  38
activities for OPERATIONS attribute  609
activities for SPECIAL attribute  620
RACF command usage by a user  177
real data set names  619
system-wide command violations  592
system-wide for RACF classes  589
logon procedure for TSO
  changing  175
  defining  85
Lotus Notes for z/OS
  general resource class  664

M
MAIN operand
  TARGET command  637
mainframe
  education xvi
management class for DFP
  changing  113
  in user profile  136
  defining  27
  for group profile  59
MAP function
  RACDCERT command  365
  RACMAP command  393
MASK operand
  SEARCH command  563
maximum number of users in group  188
maximum TSO region size
  for existing user  174
  for new user  84
MAXSIZE suboperand
  ADDUSER command  84
  ALTUSER command  174
MAXTKTLFE operand
  ADDUSER command  62
  ALTER command  437
  RDEFINE command  496
  RREPLACE command  499
MAXTKTLFE suboperand
  ALTUSER command  141
MCICSPPT class
  description  662
MDSNPK class
  description  663
MDSNPN class
  description  663
MDSNSC class
  description  663
MDSNSG class
  description  663
MDSNSM class
  description  663
MDSNSP class
  description  663
MDSNSQ class
  description  663
MDSNTB class
  description  663
MDSNTS class
  description  663
MDSNUF class
  description  663
MDSNUT class
  description  663
member
  adding to resource group  409, 467
  deleting from resource group  411
MEMLIMIT suboperand
  ADDUSER command  69
  ALTUSER command  151
message
  notify when profile denies access
    for existing data set profile  102
    for new data set profile  42
    for new general resource profile  497
  password expiration  614
  password phrase expiration  614
warning
  changing for data set profile  105
  defining for data set profile  45
  for existing general resource profile  454
  for new general resource profile  508
message class for TSO
  changing for user  175
  defining for user  85
message routing codes
  for new user profile  76
  for user profile  160
MFORM suboperand
  ADDUSER command  75
  ALTUSER command  158
MGMTCLAS class
  description  665
MGMTCLAS suboperand
  ADDGROUP command  27
  ADDUSER command  59
  ALTGROUP command  113
  ALTUSER command  136
  TARGET command  642
MIGID suboperand
  ADDUSER command  75
  ALTUSER command  159
migration id assignment
  for user profile 159
migration ID assignment
  for new user profile 75
MIMS class
description 664
MINCHANGE suboperand
  PASSWORD operand 610
minimum TSO region size
  for existing user 176
  for new user 85
MINTKTLFE operand
  RALTER command 438
  RDEFINE command 496
MIXCASE suboperand
  PASSWORD operand 611
MMPAREAMAX suboperand
ADDUSER command 69
ALTUSER command 152
MMSLSTxx PARMLIB member
  relation to SETROPTS LANGUAGE setting 603
model data set profile
  authorization required to specify 35
  copying fields from 35
defining 41
  displaying name 227
  for existing user 143
  for group data sets 114
  for new user 63
  locating volume for 40
  model for group data sets 28
  searching for 558
  system-wide processing options 608
  using existing profile as model 39
model general resource profile
  using existing profile as 491
  using volume to locate 492
MODEL operand
ADDGROUP command 28
ADDSD command 41
ADDUSER command 63
ALTGROUP command 114
ALTUSER command 143
SEARCH command 558
SETROPTS command 608
MONITOR suboperand
  ADDUSER command 75
  ALTUSER command 159
MQADMIN class
description 665
MQCMDS class
description 665
MQCONN class
description 665
MQLIST class
description 665
MQPROC class
description 665
MQQUEUE class
description 665
MQSeries
general resource classes 664
MSCOPE suboperand
  ADDUSER command 75
  ALTUSER command 160
MSGCLASS suboperand
  ADDUSER command 85
  ALTUSER command 175
MSGRECVR suboperand
  ADDUSER command 65
  ALTUSER command 145
MXADMIN
description 666
MXNLIST
description 666
MXPROC
description 666
MXQUEUE
description 666
MXTOPIC
description 666

N
NAME operand
  ADDUSER command 64
  ALTUSER command 143
NCICSPPT class
description 662
NDS operand
  ADDUSER command 64
  ALTUSER command 143
  LISTUSER command 243
NDS segment
  changing
    in user profile 143
    user profile
displaying 243
NDSLINK class
description 664
NETCMDS class
description 665
NETSPAN class
description 665
NetView
general resource classes 665
NETVIEW operand
ADDUSER command 64
ALTUSER command 144
LISTUSER command 244
NETVIEW segment
ADDDOMAINS suboperand 145
ADDOCLASS suboperand 146
changing
  in user profile 144
  CONSNAME suboperand 144
  CTL suboperand 144
  DELDOMAINS suboperand 145
  DELOCLASS suboperand 146
  DOMAINS suboperand 145
NETVIEW segment (continued)

group profile
displaying 244
IC suboperand 145
MSGRECVR suboperand 145
NGMFADMN suboperand 146
NOCONSNAME suboperand 144
NOCTL suboperand 145
NODOMAINS suboperand 145
NOIC suboperand 145
NOMSGRECVR suboperand 146
NONGMFADMN suboperand 146
NOOPCLASS suboperand 146
OPCLASS suboperand 146
new group
defining 24
new password
specifying 260
new password phrase
specifying 260
new user
defining 48
NEWLABEL operand
RACDCERT ALTER command 299
RACDCERT ALTMAP command 301
RACDCERT ROLLOVER command 381
NGMFADMN suboperand
ADDUSER command 65
ALTUSER command 146
NGMFVSPN suboperand
ADDUSER command 65
ALTUSER command 146
no security label for TSO
for existing user 175
NOACCTNUM suboperand
ALTUSER command 173
NOADDCREATOR operand
SETROPTS command 588
NOADSP operand
ADDUSER command 53
ALTUSER command 128
CONNECT command 188
SETROPTS command 588
NOALTGRP suboperand
ALTUSER command 155
NOAPPLAUDIT operand
SETROPTS command 589
NOAPPLDATA operand
RALTER command 414
NOASSIZEMAX suboperand
ALTUSER command 147
NOAUDIT operand
SETROPTS command 589
NOAUDITOR operand
ADDUSER command 54
ALTUSER command 128
CONNECT command 189
NOAUTH suboperand
ALTUSER command 156
NOAUTO suboperand
ALTUSER command 156
NOAUTODIRECT operand
SET command 572
NOBINDDDN suboperand
ALTUSER command 167
RALTER command 441
NOBINDPW suboperand
ALTUSER command 168
RALTER command 442
NOCICS operand
ALTUSER command 131
NOCLASSACT operand
RVARY command 549
SETROPTS command 591
NOCLAUTH operand
ADDUSER command 56
ALTUSER command 131
NOCMDSYS suboperand
ALTUSER command 156
NOCMDVIOL operand
SETROPTS command 592
NOCOMMAND suboperand
ALTUSER command 173
NOCOMPATMODE operand
SETROPTS command 592
NOCONSNAME suboperand
ALTUSER command 144
NOCPUTIME MAX suboperand
ALTUSER command 150
NOCSDATA operand
ALTGROUP command 112
ALTUSER command 132
NOCOMMAND suboperand
ALTUSER command 173
NODE operand
ALTDS command 99
ALTGROUP command 112
ALTUSER command 132
RALTER command 428
NODATAAPP operand
ALTGROUP command 112
ALTUSER command 136
NODATAclass operand
ALTGROUP command 113
ALTUSER command 136
NODATASHARE operand
RVARY command 550
NODEFTKTLFE operand
TARGET command 637
NODMBR class
description 660
NODGROUP operand
ALTUSER command 174
NODFP operand
ALTGROUP command 113
ALTUSER command 137
NODMBR class
description 660
NODOM suboperand
ALTUSER command 157
NODOMAINS suboperand
  ALTUSER command 145
NOEGN operand
  SETROPTS command 593
NOEIM operand
  ALTUSER command 137
NOENCRYPT operand
  RALTER command 436
NOENCRYPT suboperand
  ALTUSER command 139
NOERASE operand
  ALTDSD command 100
  SETROPTS command 594
NOEXPIRED operand
  ALTUSER command 138
NOFILEPROCMAp suboperand
  ALTUSER command 150
NOGENCMMD operand
  SETROPTS command 595
NOGENERIC operand
  LISTDSD command 220
  RLST command 532
  SEARCH command 558
  SETROPTS command 597
NOGENERICOWNER operand
  SETROPTS command 598
NOGENLIST operand
  SETROPTS command 599
NOID suboperand
  ALTGROUP command 116
NOGLOBAL operand
  SETROPTS command 600
NOGROUP operand
  RALTER command 449
NOGRPPACC operand
  ADDUSER command 60
  ALTUSER command 138
  CONNECT command 189
NOGRPLIST operand
  SETROPTS command 601
NOHC suboperand
  ALTUSER command 157
NOHISTORY suboperand
  PASSWORD operand 610
NOHOLDCLASS suboperand
  ALTUSER command 174
NOHOME suboperand
  ALTUSER command 151
NOIC suboperand
  ALTUSER command 145
NOINACTIVE operand
  SETROPTS command 601
NOINITSTATS operand
  SETROPTS command 601
NOINTERVAL operand
  PASSWORD operand 260
  PHRASE command 260
NOINTIDS suboperand
  ALTUSER command 157
NOJOBCLASS suboperand
  ALTUSER command 174
NOKERB operand
  ALTUSER command 141
NOKERBNNAME operand
  RALTER command 437
NOKERBNNAME suboperand
  ALTUSER command 141
NOKEY suboperand
  ALTUSER command 157
NOLANGUAGE operand
  ALTUSER command 142
NOLDAPHOST suboperand
  ALTUSER command 167
  RALTER command 441
NOLEVEL suboperand
  ALTUSER command 158
NOLIST operand
  RVARY command 552
  SEARCH command 563
NOLOGCMDRESP suboperand
  ALTUSER command 158
NOMASK operand
  SEARCH command 563
NOMAXSIZE suboperand
  ALTUSER command 175
NOMAXXTLFE operand
  RALTER command 438
NOMAXXTLFE suboperand
  ALTUSER command 141
NOMEMLIMIT suboperand
  ADDUSER command 69
  ALTUSER command 151
NOMFORM suboperand
  ALTUSER command 159
NOMGMTCLAS suboperand
  ALTGROUP command 113
ALTMEDUSER suboperand
  ALTUSER command 136
NOMIGID suboperand
  ALTUSER command 159
NOMINTKLFE operand
  RALTER command 438
NOMIXEDCASE suboperand
  PASSWORD operand 611
NOMMAPAREAMAX suboperand
  ALTUSER command 152
NOMODEL operand
  ALTGROUP command 114
  ALTUSER command 143
  SETROPTS command 609
NOMONITOR suboperand
  ALTUSER command 159
NOMSCPOE suboperand
  ALTUSER command 160
NOMSGCLASS suboperand
  ALTUSER command 175
NOMSGRECVR suboperand
  ALTUSER command 146
  nonautomatic TAPEVOL profile altering 401
  defining 508
  permitting access to 264
NONGMFADMN suboperand
   ALTUSER command 146
NONGMFVSPN suboperand
   ALTUSER command 146
NONOTIFY operand
   ALTDSD command 102
   RALTER command 440
NONOTIFY suboperand
   SET command 572
NONVSM operand
   SEARCH command 559
NOOIDCARD operand
   ADDUSER command 66
   ALTUSER command 147
NOOMVS operand
   ALTGROUP command 116
NOOMVS suboperand
   ALTUSER command 154
NOOPCLASS suboperand
   ALTUSER command 129, 146
NOOPERATIONS operand
   ADDUSER command 72
   ALTUSER command 154
   CONNECT command 190
NOOPERAUDIT operand
   SETROPTS command 609
NOOPERPARAM operand
   ALTUSER command 161
NOOPIDENT suboperand
   ALTUSER command 129
NOOPPRTY suboperand
   ALTUSER command 129
NOOUTPUT suboperand
   SET command 572
NOOVM operand
   ALTGROUP command 117
NOPADCHK suboperand
   RDEFINE command 473
NOPASSWORD operand
   ADDUSER command 78
   ALTUSER command 165
   RALTER command 439
NOPHRASE operand
   ALTUSER command 166
NOPREFIX operand
   SETROPTS command 615
NOPERPRIMARY suboperand
   ALTUSER command 142
NOPERIVILEGED operand
   RALTER command 449
NOPROC suboperand
   ALTUSER command 175
NOPROCMAX suboperand
   ALTUSER command 153
NOPROGRAM suboperand
   ALTUSER command 154
NPROTECTALL operand
   SETROPTS command 616
NPROXY operand
   ALTUSER command 168
   RALTER command 442
NORACF operand
   LISTGRP command 230
   LISTUSER command 244
NORACLIST operand
   SETROPTS command 619
NOREALDSN operand
   SETROPTS command 620
NORESTRICTED operand
   ADDUSER command 82
   ALTUSER command 169
NORESUME operand
   ALTUSER command 170
   CONNECT command 191
NOREVOKE operand
   ALTUSER command 171
   CONNECT command 192
NOREVOKE suboperand
   PASSWORD operand 612
NOROUTCODE suboperand
   ALTUSER command 161
NORSKEY suboperand
   ALTUSER command 130
NORULEn suboperand
   PASSwort operand 614
NORULES suboperand
   PASSWORD operand 614
NOSAUDIT operand
   SETROPTS command 621
NOSACCOUNT operand
   ALTDSD command 103
   ALTUSER command 171
   RALTER command 442
NOSCLALabel operand
   ALTUSER command 175
NOSCLALBALAUDIT operand
   SETROPTS command 621
NOSINDEX operand
   SETROPTS command 621
NOSLEVEL operand
   ALTUSER command 172
   RALTER command 443
NOSLEVELAUDIT operand
   SETROPTS command 622
NOSSECONDARY suboperand
   ALTUSER command 142
NOSSESSION operand
   RALTER command 444
NOSET operand
   ADDSD command 41
   ALTUSER command 172
   DELDSD command 197
NOSHMEMMAX suboperand
   ADDUSER command 71
NOSIgVER operand
   ALTUSER command 153
NOSIGVER operand
   ALTUSER command 142
NOSINGLEDSN operand
   RALTER command 447
NOSSECLEVEL operand
   ALTUSER command 175
   RALTER command 447
NOSSECONDARY suboperand
   ALTUSER command 142
NOSNAME suboperand
   ALTUSER command 176
NOSNAME suboperand
   ALTUSER command 143
NOSPECIAL operand
   ADDUSER command 83
   ALTUSER command 172
   CONNECT command 192
NOSTATISTICS operand
   SETROPTS command 623
NOSTORAGE suboperand
   ALTUSER command 161
NOSTORCLAS suboperand
   ALTGROUP command 113
   ALTUSER command 176
NOTAFTER(DATE) operand
   RACDCERT GENCERT command 334
   RACDCERT REKEY command 373
NOTAPE suboperand
   RVARY command 549
NOTAPEDSN operand
   SETROPTS command 623
NOTBEFORE(DATE) operand
   RACDCERT GENCERT command 334
   RACDCERT REKEY command 373
NOTELINK class
   description 664
NOTERMUACC operand
   ADDGROUP command 30
   ALTGROUP command 117
NOTTHREADSMAX suboperand
   ALTUSER command 153
Notices 671
NOTIFY operand
   ADDSD command 42
   ALTSD command 102
   RALTER command 439
   RDEFINE command 497
NOTIFY suboperand
   SET command 570
NOTIMEOUT suboperand
   ALTUSER command 130
   NOTIMEZONE operand
   RALTER command 451
NOTRACE operand
   RALTER command 449
NOTRUST operand 298
   RACDCERT ADD command 288
   RACDCERT ALTMAP command 302
   RACDCERT MAP command 369
NOTRUSTED operand
   RALTER command 449
NOTSLSKEY suboperand
   ALTUSER command 131
NOTSO operand
   ALTUSER command 177
NOTVTOC operand
   RALTER command 453
NOUAUDIT operand
   ALTUSER command 178
NOUD suboperand
   ALTUSER command 161
NOUID suboperand
   ALTUSER command 149
NOTAFTER operand
   ALTUSER command 161
NOTAPEDSN operand
   ALTUSER command 177
Novell Directory Services for OS/390
   general resource class 664
NOWARNING operand
   ALTUSER command 105
   RALTER command 454
NOWARNING suboperand
   RALTER command 454
NOWHEN(PROGRAM) operand
   SETROPTS command 624
NOXRFSOFF suboperand
   ALTUSER command 131
NOYOURACC operand
   RLIST command 534
NVASAPDT class
   description 665
O
OIDCARD (operator identification card)
   See operator identification card (OIDCARD)
OIDCARD operand
   ADDUSER command 66
   ALTUSER command 146
OIMS class
   description 664
OMVS operand
   ADDGROUP command 28
   ADDUSER command 66
   ALTGROUP command 114
   ALTUSER command 147
   LISTGRP command 230
   LISTUSER command 244
OMVS profile
   OMVS for group data sets 28
OMVS segment
   changing
      in group profile 114
      in user profile 147
   defining
      for new user profile 66
      group profile displaying 230
      user profile displaying 244
ONLYAT operand
   ADDGROUP command 26
   ADDSD command 38
ONLYAT operand (continued)
ADDUSER command 54
ALTDS command 99
ALTGRO command 111
ALTUSER command 128
CONNECT command 188
DELDSD command 196
DELGROUP command 201
DELUER command 204
LISTDS command 218
LISTGRP command 230
LISTUSER command 242
PASSWORD command 259
PERMIT command 267
PHRASE command 259
RALTER command 414
RDEFINE command 476
RDELETE command 517
REMOVE command 521
RLIST command 531
SEQU command 559
SETROPTS command 589
OPCLASS operand
ADDUSER command 65
ALTUSER command 129, 146
OPERANDS operand
HELP command 212
OPERATIONS attribute
for existing user 154
for new user 72
logging activities for 609
OPERATIONS operand
ADDUSER command 72
ALTUSER command 154
CONNECT command 189
OPERATIVE operand
TARGET command 635
operator identification card (OIDCARD)
for existing user 146
for new user 66
operator information
changing command authority 155
delete from profile 161
for user profile 155
OPERAUDIT operand
SETROPTS command 609
OPERCMD operand
description 660
OPERPAM operand
ADDUSER command 72
ALTUSER command 155
LISTUSER command 245
OPERATION segment
alter operator command authority
for user profile 155
AUTH suboperand 155
AUTO request reception
delete from user profile 156
AUTO suboperand 73, 156
CMDSYS suboperand 73, 156
command response logging
for new user profile 74
for user profile 158
console command system
for new user profile 73
for user profile 156
console message format
for new user profile 75
for user profile 158
console message storage
for new user profile 76
for user profile 161
console operator command authority
for new user 73
console search key
for new user profile 74
for user profile 157
deleting
console command system from user profile 156
console search key from user profile 157
from profile 161
message level from user profile 158
operator command authority from user profile 156
displaying for user profile 245
DOM request reception
deleting from user profile 157
for new user profile 73
for user profile 156
DOM suboperand 73, 156
event display information
for new user profile 75
for user profile 159
HC suboperand 74, 157
INTIDS suboperand 74, 157
KEY suboperand 74, 157
LEVEL suboperand 74, 157
LOGCMRESP suboperand 74, 158
message routing codes
for new user profile 76
for user profile 160
MFORM suboperand 75, 158
MIGID suboperand 75, 159
migration id assignment
for user profile 159
migration ID assignment
for new user profile 75
MOTION suboperand 75, 159
MSCOPE suboperand 75, 160
NOAUTH suboperand 156
NOAUTO suboperand 156
NOCMDSYS suboperand 156
NODOM suboperand 157
NOHIC suboperand 157
NOINTIDS suboperand 157
NOKEY suboperand 157
NOLEVEL suboperand 158
NOLOGCMDRESP suboperand 158
NOMFORM suboperand 159
NOMIGID suboperand 159
OPERPARM segment (continued)
  NOMONITOR suboperand 159
  NOMSCOPE suboperand 160
  NOROUTCODE suboperand 161
  NOSTORAGE suboperand 161
  NOUD suboperand 161
  NOUNKNIDS suboperand 161
  ROUTCODE suboperand 76, 160
  STORAGE suboperand 76, 161

  system message reception
    for new user profile 75
    for user profile 160

  type of broadcast messages
    for new user profile 74
    for user profile 157

  UD suboperand 76, 161

  undelivered message reception
    for new user profile 76
    for user profile 161

  UNKNIDS suboperand 76, 161

  user profile
    for new user 72

OPIDENT suboperand
  ALTUSER command 129

OPPRTY suboperand
  ALTUSER command 129

  options
    displaying current RACF 603
    setting system-wide RACF 582

OUTPUT suboperand
  SET command 572

OVM operand
  ADDGROUP command 29
  ADDUSER command 77
  ALTGROUP command 116
  LISTUSER command 245

OVM profile
  OVM for group data sets 29

OVM segment
  changing
    in group profile 116
    displaying for user profile 245

OVM suboperand
  ALTUSER command 162

OWNER operand
  ADDGROUP command 30
  ADDSD command 42
  ADDUSER command 78
  ALTGROUP command 117
  ALTUSER command 164
  CONNECT command 190
  RALTER command 440
  RDEFINE command 498
  REMOVE command 521

P

PADCHK suboperand
  RDEFINE command 473
PERMDIR 1
PERMFILE 1
PERMIT command
  authorization required 265
description 263
text 273
syntax 265
permitting access to profiles 265
persistent verification 206, 501, 628
PHRASE command
  authorization required 258
description 257
text 261
text 259
PHRASE operand
  ADDUSER command 79
  ALTUSER command 165
  PASSWORD command 260
  PHRASE command 260
PIMS class
description 664
PMBR class
description 660
POE operand
  DISPLAY command 207
  SIGNOFF command 629
PREFIX operand
  LISTDSD command 218
  SETROPTS command 615
  TARGET command 637
preventing
  access to profiles 265
  user from accessing system 170, 171, 191, 192
PRIMARY suboperand
  ALTUSER command 142
Print Services Facility (PSF) for z/OS
  PSFMPL, general resource class 661
PRINTSERV class
description 664
PRIVILEGED operand
  RALTER command 449
  RDEFINE command 505
PROC suboperand
  ADDUSER command 85
  ALTUSER command 157
PROAC class
description 666
PROCESS class
description 666
PROCUSER suboperand
  ADDUSER command 70
PROCUSERMAX suboperand
  ALTUSER command 152
profile
  defining with enhanced generic naming 650
  locating in the RACF database 555
profile name (continued)
data set profile (continued)
  displaying 218
  using as model 39
general resource profile
  changing 407
  defining 465
  deleting 516
  displaying 530
  using for model profile 491
  modifying access list for 266
syntax 11
PROGRAM class
description 660
program control
  activating 623
  conditional access list 263, 272
  controlled program
  access for 272
  defining 408, 466
  deactivating 623
  in-storage table
  changing 472
  refreshing 620
PROGRAM suboperand
  ADDUSER command 70, 77
  ALTUSER command 154
PROPCNTL class
description 660
PROTECTALL operand
  SETROPTS command 615
PROTECTALL processing
  activating or deactivating 615
  protected user ID 79, 165
PROTOCOL operand
  TARGET command 638
PROXY operand
  ADDUSER command 80
  ALTUSER command 166
  LISTUSER command 245
  RALTER command 440
  RDEFINE command 498
  RLIST command 534
PROXY segment
  displaying for user profile 245
PSFMPL class
description 661, 667
PTKTDATA class
description 661, 667
PTKTVAL class
description 665, 667
publications
  on CD-ROM and DVD xiii
  softcopy xiii
PURGE operand
  TARGET command 639
PWSYNC operand
  SET command 574
Q
QCICSPSB class
description 662
QIMS class
description 664

R
RACDCERT ADD command
authorization required 286
description 283
examples 292
syntax 287
RACDCERT ADDRING command
authorization required 293
description 293
examples 294
syntax 293
RACDCERT ADDTOKEN command
authorization required 295
description 295
examples 296
syntax 295
RACDCERT ALTER command 298
authorization required 297
description 297
examples 299
syntax 298
RACDCERT ALTMAP command
authorization required 300
description 300
examples 302
syntax 301
RACDCERT BIND command
authorization required 303
description 303
examples 306
syntax 304
RACDCERT CHECKCERT command
authorization required 307
description 307
examples 309
syntax 307
RACDCERT command, common
authorization required 277
description 276
examples 280
list of functions 276
RACDCERT command, general purpose 276
RACDCERT CONNECT command
authorization required 310
description 310
examples 313
syntax 311
RACDCERT DELETE command
authorization required 314
description 314
examples 316
syntax 315
RACDCERT DELMAP command
authorization required 317
description 317
examples 319
syntax 318
RACDCERT DELRING command
authorization required 320
description 320
examples 321
syntax 320
RACDCERT DELTOKEN command
authorization required 322
description 322
examples 323
syntax 322
RACDCERT EXPORT command
authorization required 324
description 324
examples 327
syntax 325
RACDCERT GENCERT command
authorization required 330
description 328
examples 339
syntax 331
RACDCERT GENREQ command
authorization required 341
description 340
examples 342
syntax 341
RACDCERT IMPORT command
authorization required 343
description 343
examples 347
syntax 344
RACDCERT LIST command
authorization required 349
description 348
examples 351
syntax 349
RACDCERT LISTMAP command
authorization required 355
description 355
examples 356
syntax 356
RACDCERT LISTRING command
authorization required 358
description 358
examples 359
syntax 358
RACDCERT LISTTOKEN command
authorization required 364
description 364
examples 370
syntax 365
RACDCERT REKEY command
authorization required 371
description 371
examples 376
syntax 372
RACDCERT REMOVE command
authorization required 377
description 377
examples 378
syntax 378
RACDCERT ROLLOVER command
authorization required 380
description 379
examples 381
syntax 380
RACDCERT UNBIND command
authorization required 382
description 382
examples 384
syntax 382
RACF
publications
on CD-ROM and DVD xiii
softcopy xiii
RACF commands
basic information for issuing RACF Commands 7
descriptions 15
logging usage by a user 177
obtaining help for 211
operator commands 21
return codes 11
summary of 1
symbols used in syntax diagrams 11
syntax 9
RACF indication
for existing data set profile 101
for new data set profile 41
RACF operator commands
descriptions 21
entering 21
RACF options
displaying current 603
example of display 627
RACF protection
removing from data set profile 194
RACF resource protection
deactivating or reactivating
using RVARY command 546
RACF segment
group profile
changing 108
defining 24
displaying 230
suppressing display 230
user profile
changing 120
defining 49
displaying 241
suppressing display 244
RACF TSO commands (continued)
entering
background 18
foreground 17
RACFEVNT class
description 661
RACFHC class
description 661
RACFVARS class
description 661, 667
RACGLIST class
description 661
RACHCMBR class
description 661
RACLINK command
authorization required 385, 399
description 385
examples 390
syntax 386
RACLIST operand
SETROPTS command 616
RACMAP command
authorization required 392
description 392
examples 397
syntax 393
RACPRIV command
description 399
examples 400
syntax 399
RALTER command
authorization required 402
description 401
examples 456
syntax 403
RAUDITX class
description 663
RCICSRES class
description 662
RDATALIB class
description 661
RDEFINE command
authorization required 460
description 459
examples 510
syntax 462
RDELETE command
authorization required 515
description 514
examples 517
syntax 515
reactivating
RACF resource protection
using RVARY command 546
REALDSN operand
SETROPTS command 619
REALM class
description 665
recording
RACROUTE REQUEST=VERIFY statistics 601
real data set names 619
recording (continued)  622
REFRESH operand  620
refreshing  620
  generic profile checking  596
  global access checking  600
  in-storage generic profiles and program control tables  620
region size for TSO  610
  maximum  610
    for existing user  174
    for new user  84
  minimum  609
    defining for user  85
    for existing user  176
REKEY function  611
RACDCERT command  372
REMOVE command  519
  authorization required  519
  description  519
  examples  521
  syntax  520
REMOVE function  378
RIMS class  664
  description  664
RING operand  312
RROTFDATA class  665
  description  665
ROLES operand  665
  description  665
ROLLOVER function  381
ROUTCODE suboperand  76
RULE suboperand  612
  establishing password syntax  17
RVARSMBR class  661
  description  661, 667
RVARY command  552
  authorization required  548
  description  546
  examples  552
  syntax  548
RVARYPW operand
   SETROPTS command 620

S
   sample
      ISPF panel 17
   SAUDIT operand
      SETROPTS command 620
   SCDMBR class
      description 661, 667
   SCICSTST class
      description 662
   SDNFILTER operand
      RACDCERT MAP command 367
   SDSF (System Display and Search Facility)
      general resource class 661
   SDSF class
      description 661
   SEARCH command
      authorization required 556
      description 555
      examples 565
      syntax 557
   searching for profiles in RACF database 555
   SECDATA class
      description 661, 667
   SECLABEL class
      description 661, 667
   SECLABEL operand
      ADDSD command 43
      ADDUSER command 82
      ALTSDSD command 103
      ALTUSER command 171
      DISPLAY command 208
      RALTER command 442
      RDEFINE command 500
      SEARCH command 559
      SIGNOFF command 630
   SECLABEL suboperand
      ADDUSER command 85
   SECLABELAUDIT operand
      SETROPTS command 621
   SECLEVEL operand
      ADDSD command 43
      ADDUSER command 82
      ALTSDSD command 103
      ALTUSER command 171
      DISPLAY command 208
      RALTER command 442
      RDEFINE command 500
      SEARCH command 559
   SECLEVELAUDIT operand
      SETROPTS command 622
   SECLMBR class
      description 661
   SECONDARY suboperand
      ALTUSER command 142
   security category (continued)
      changing (continued)
         general resource profile 408
      defining 471
         for data set profile 37
         for general resource profile 467
         for user profile 53, 66, 68, 69, 70, 71
      deleting 411
         from general resource profile 409
         from user profile 127
      undefined category names 409
   security classification of users and data
      defining categories and levels 471
      in data set profile 38, 98
      in general resource profile 409, 467
      in user profile 53, 66, 68, 70, 72, 127
   security label
      changing
         for user profile 171
      defining
         for user profile 82
         for new user 85
         for TSO 85
      for user profile 43, 500
      translating on inbound jobs or SYSOUT 472
   security level
      changing
         for data set profile 103
      defining 471
      deleting 411
      for existing data set profile 103
      for existing general resource profile 442
      for existing user profile 171
      for new data set profile 43
      for new general resource profile 500
      for new user profile 82
      for resource profile 442
      in user profile 171
      logging access attempts based on 622
   search on general resource profiles 559
      using as search criteria 559
   security retention period
      for existing tape data set profile 103
      for new tape data set profile 43
      system-wide for tape data sets 620
   security topics for RACF
      classroom courses  xiv
   segment
   CICS segment
      displaying for a user profile 242
      for new user profile 54
   CSDATA segment
      displaying for group profile 230
      displaying for user profile 242
      for existing group profile 111
      for existing user profile 131
      for new group profile 26
      for new user profile 56
segment (continued)
DFP segment
  changing in group profile 112
  changing in user profile 135
  displaying for a user profile 243
  displaying for group profile 230
  existing user profile, deleting from 137
  for new group profile 27
  for new user profile 59
EIM segment
  displaying for a user profile 243
KERB segment
  displaying for a user profile 243
LANGUAGE segment
  displaying for a user profile 243
LNOTES segment
  changing in user profile 142
  displaying for a user profile 243
NDS segment
  changing in user profile 143
  displaying for a user profile 243
NETVIEW segment
  changing in user profile 144
  displaying for group profile 244
OMVS segment
  changing in group profile 114
  changing in user profile 147
  displaying for group profile 230
  displaying for user profile 244
  for new user profile 66
OPERPARM segment
  displaying for user profile 245
  for new user profile 72
OVM segment
  changing in group profile 116
  displaying for user profile 245
PROXY segment
  displaying for a user profile 245
RACF segment
  changing for a group profile 108
  defining for group profile 24
  displaying for group profile 230
  displaying for user profile 241
  for existing user profile 120
  for new user profile 49
  suppressing display for group profile 230
  suppressing display for user profile 244
SESSION segment
  displaying for general resource profile 535
  for general resource profiles 501
SIGVER segment
  displaying for general resource profile 535
  for general resource profiles 502
SSIGNON segment
  displaying 535
STDATA segment
  defining 505
  modifying 448
TSO segment
  deleting from user profile 177
  displaying for a user profile 246

segment (continued)
TSO segment (continued)
  for existing user profile 172
  for new user profile 83
WORKATTR segment
  displaying for group profile 246
SERVAUTH class
  description 661
SERVER class
  description 661
SESSION operand
  RALTER command 443
  RDEFINE command 501
  RLIST command 535
SESSION segment
  displaying
    general resource profile 535
SET command
  authorization required 567
  callable service trace types
    IBM Policy Director 577
    RACROUTE 578
    z/OS UNIX 576
  description 567
  examples 579
  syntax 567
SET operand
  ADDSD command 41
  ALTDS command 101
  DELDS command 197
SETONLY operand
  ADDSD command 41
SETROPTS command
  authorization required 584
  description 582
  examples 624
  syntax 585
SFSCMD class
  description 667
  shared file system profiles 657
  shared in-storage profile
    SETROPTS GENLIST processing for 598
    SETROPTS RACLIST processing for 616
SHARED suboperand
  ALTGROUP command
    for GID suboperand 29, 115
  UID suboperand
    ADDUSER command 67
    ALTUSER command 149
SHMEMMAX suboperand
  ADDUSER command 71
  ALTUSER command 153
  shortcut keys 669
  signed-on-from list 206, 628
SIGNOFF command
  authorization required 628
  description 628
  examples 630
  syntax 629
SIGNON operand
  DISPLAY command 207
SIGNWITH operand
   RACDCERT GENCERT command 335

SIGVER operand
   RALTER command 444
   RDEFINE command 502
   RLIST command 535

SIGVER segment
   displaying
      general resource profile 535

SIMS class
   description 664

SINGEDSN operand
   RALTER command 447
   RDEFINE command 504

SIZE operand
   RACDCERT GENCERT command 333
   RACDCERT REKEY command 373

SIZE suboperand
   ADDUSER command 85
   ALTUSER command 176

SMESSAGE class
   description 661

SMS (Storage Management Subsystems)
   general resource classes 665

SNAME suboperand
   ALTUSER command 142

SOMDOJS class
   description 661

SPECIAL attribute
   for existing user 172
   for new user 83
   logging activities for 620

SPECIAL operand
   ADDUSER command 83
   ALTUSER command 172
   CONNECT command 192

specifying the distinguished name the LDAP proxy
server will use
   ADDUSER command 80
   ALTUSER command 167
   RALTER command 441
   RDEFINE command 499

specifying the password the LDAP proxy server will use
   ADDUSER command 81
   ALTUSER command 168
   RALTER command 441
   RDEFINE command 499

specifying the URL of the LDAP proxy server
   ADDUSER command 80
   ALTUSER command 166
   RALTER command 440
   RDEFINE command 498

SRDIR 1
SRFILE 1

SSIGNON operand
   SSIGNON segment
      displaying
         general resource profile 535
   standard access list 263
   STARTED class
      description 661
      started task
         security category checking
            RALTER command 408
            RDEFINE command 467
         security level checking 443
      statistics
         bypassing recording of 601
         displaying
            for data set profile 221
            general resource profile 535
         recording
            for classes 622
            for REQUEST=VERIFY processing 601

STATISTICS operand
   LSTDSD command 221
   RLIST command 535
   SETROPTS command 622

STATUS suboperand
   RVARYPW operand (SETROPTS command) 620

STDATA operand
   RALTER command 448
   RDEFINE command 505
   RLIST command 536

STDATA segment
   displaying
      general resource profile 536
   RALTER command 448
   RDEFINE command 505

STOP command
   authorization required 631
   description 631
   examples 632
   syntax 631

storage class for DFP
   adding
      to new group profile 27
   changing
      for group profile 113
      in user profile 136
   defining
      for user profile 60

STORAGE suboperand
   ADDUSER command 76
   ALTUSER command 161

STORCLAS class
   description 665

STORCLAS suboperand
   ADDGROUP command 27
   ADDUSER command 60
   ALTGROUP command 113
   ALTUSER command 136
   TARGET command 642

SUBJECTSDN operand
   RACDCERT GENCERT command 333
SUBSYSNM class
  description 665
subsystem prefix
  ADDBGROUP command 26
  ADDSD command 36
  ADDUSER command 53
  ALTSDS command 96
  ALTGROUP command 110
  ALTUSER command 127
  CONNECT command 187
  DELSDS command 196
  DELGROUP command 200
  DELUSER command 204
  DISPLAY command 207
  LISTSDS command 217
  LISTGRUP command 229
  LISTUSER command 241
  PASSWORD command 259
  PERMIT command 266
  PHRASE command 259
  RACLNNK command 387
  RALTER command 407
  RDEFINE command 465
  REMOVE command 520
  RESTART command 523
  RLIST command 529
  RVARY command 548
  SEARCH command 558
  SET command 568
  SETROPTS command 587
  SIGNOFF command 629
  STOP command 632
  TARGET command 634

summary
  of RACF authorities 1
  of RACF commands 1
superior group
  for existing group 117
  for new group 30
SUPGROUP operand
  ADDBGROUP command 30
  ALTGROUP command 117
suppressing display of RACF segment
  group profile 230
  user profile 244
SURROGAT class
  description 661
SVFMR operand
  RALTER command 449
  RDEFINE command 506
  RLIST command 536
SWITCH operand
  RVARY command 550
SWITCH suboperand
  RVARYPW operand (SETROPTS command) 620
SYNTAX operand
  HELP command 212
syntax rules
  for commands 9
  for passwords 612
SYS suboperand
  ADDUSER command 86
  ALTUSER command 176
SYSMVIEW class
  description 661
SYSNAME operand
  TARGET command 639
SYSOUT class for TSO
  for existing user 176
  for new user 86
SYSOUT data set destination
  for existing user 173
  for new user 84
systplex communication
  data sharing option 5
  example of RACGLIST processing 511
  example of RVARY NODATASHARE 552
  logging RVARY commands 546
  RVARY ACTIVE command 549
  RVARY INACTIVE command 549
  RVARY SWITCH command 551
System Display and Search Facility (SDSF)
  See SDSF (System Display and Search Facility)
system message reception
  for new user profile 75
  for user profile 160
system-wide options
  activating 582
  displaying current RACF 603
  example of display 627
T
  tape data set
    creating entry in TVTOC 41
    defining a new profile to protect 41
    erase-on-scratch processing
      activating 593
      deactivating system-wide 594
    file sequence number 39
    searching for profile 558
    security retention period for existing profile 103
    security retention period for new profile 43
    specifying tape volume to contain single data set 447, 504
  system-wide security retention period 620
tape data set protection
  activating or deactivating 623
TAPE operand
  ADDSD command 41
  SEARCH command 558
tape volume
  creating TVTOC for 453, 508
  deactivating protection 549
  displaying volume information for 564
  searching for expired 559
  specifying to contain single data set 447, 504
TAPEDSN operand
  SETROPTS command 623
TAPEVOL class
  description 661, 667
TAPEVOL profile
   changing to nonautomatic  264
TARGET command
   authorization required  633
   description  633
   examples  643
   syntax  633
TCICSTRN class
   description  662
TEMPDSN class
   description  661
terminal
   limiting access to  454, 509
   time zone  450, 506
   UACC for undefined terminals  623
terminal authorization checking
   for users in a new group  30
   for users in an existing group  117
TERMINAL class
   description  661, 667
terminal id
   syntax  11
TERMINAL operand
   SETROPTS command  623
TERMUUACC operand
   ADDGROUP command  30
   ALTGROUP command  117
THREADSMAX suboperand
   ADDUSER command  71
   ALTUSER command  153
time of day
   existing user can access system  178
   new user can access system  87
   terminal can access system  455, 509
TIME operand
   ADDUSER command  87
   ALTUSER command  178
   RACDCERT GENCERT command  334
   RACDCERT REKEY command  373
   RALTER command  454
   RDEFINE command  509
TIMEOUT suboperand
   ADDUSER command  55
   ALTUSER command  130
TIMEZONE operand
   RALTER command  450
   RDEFINE command  506
TIMS class
   description  664
Tivoli
   general resource class  665
Tivoli Service Desk
   general resource classes  664
TME operand
   (continued)
   RDEFINE command  507
   RLIST command  231
TMEDADMIN class
   description  665
TRACE operand
   RALTER command  449
   RDEFINE command  505
   SET command  574
TRUST operand
   RALTER command  449
   RDEFINE command  506
TSLKEY suboperand
   ALTUSER command  130
TSO default prefix for data set  18
TSO logon information
   changing
      default for user profile  172
      defining
         default for user profile  83
      deleting
         from user profile  177
   for user profile  157
TSLKEY operand
   ADDGROUP command  30, 31
   ADDUSER command  83
   ALTUSER command  172
   LISTUSER command  246
TSO segment
   deleting from user profile  177
   displaying for user profile  246
   for existing user profile  172
   for new user profile  83
TSO SUBMIT command  19
TSO/E
   general resource classes  665
TSOAUTH class
   description  665
TSOPROC class
   description  665
TVTOC (tape volume table of contents)
   creating entry for tape data set  41
TVTOC operand
   RALTER command  453
   RDEFINE command  508
   RLIST command  536
type of broadcast messages
   for new user profile  74
   for user profile  157
U
UACC (universal access authority)
   changing
      default for user profile  177
      default in user’s connect profile  192
      for data set profile  105
      for general resource profile  454
UACC (universal access authority) (continued)
  defining
    default for user profile 86
    default in user’s connect profile 192
    for data set profile 44
    for general resource profile 508
    for undefined terminals 623
UACC operand
  ADDSD command 44
  ADDUSER command 86
  ALTUSER command 177
  CONNECT command 192
  RALTER command 454
  RDEFINE command 508
UAUDIT operand
  ALTUSER command 177
UCICSTST class
description 662
UD suboperand
  ADDUSER command 76
  ALTUSER command 177
UID operand
  SEARCH command 564
UID suboperand
  ADDUSER command 67, 78
  ALTUSER command 148
UIMS class
description 664
UNAME suboperand
  ALTUSER command 143
UNBIND function
  RACDCERT command 383
undelivered message reception
  changing for user profile 161
  defining for user profile 76
unit devices for TSO
  for existing user 176
  for new user 86
UNIT operand
  ADDSD command 45
  ALTSDSD command 105
UNIT suboperand
  ADDUSER command 86
  ALTUSER command 176
unit type
  changing for data set profile 105
  defining for data set profile 45
universal group
  for new group 31
UNIVERSAL operand
  ADDGROUP command 31
UNIXMAP class
description 666
UNIXPRIV class
description 666
UNKNIDS suboperand
  ADDUSER command 76
  ALTUSER command 161
URI operand
  RACDCERT GENCERT command 338
USAGE operand
  RACDCERT CONNECT command 312
USE group authority
description 13
user
  limiting access to system 86, 178
user data for TSO
  changing 177
  defining 86
user ID
  as new owner
    of data set profiles of removed user 521
  as owner
    of connect profile 190
    of data set profile 102
    of general resource profile 440
    of group profile 117
    of new data set profile 42
    of new general resource profile 498
    of new group profile 30
    of new user profile 78
    of user profile 164
changing access to resource for 269
deactivating an unused 601
displaying data set profiles for 218
displaying user profile for 241
no password 79, 165
protected 79, 165
removing user from group 520
revoking based on consecutive unsuccessful attempts 611
syntax 10
to add new user profile 53
to alter user profile 127
to change password 261
to change password phrase 261
to connect to group 188
to receive notify message
  for existing data set profile 102
  for general resource profile 439, 497
  for new data set profile 42
translating on inbound jobs or SYSOUT 472
when deleting user profile 204
user name
  changing 143
  defining 64
USER operand
  DISPLAY command 208
  PASSWORD command 261
  PHRASE command 261
  RALTER command
    =MEMBER operand 448
  RDEFINE command
    =MEMBER operand 505
  SEARCH command 564
  SIGNOFF command 629
user profile
  changing 120
  defining 48
deleting 203
displaying 237
user profile (continued)
  listing envelope information 238
  RACF segment 49
  removing from group 519
  searching for based on last reference 558
USERDATA suboperand
  ADDUSER command 86
  ALTUSER command 177
userid-named data set
  activating model profile for 609

V
valid values
  FORMAT operand 325
VCICSCMD class
  description 662
VMBATCH class
  description 667
VMBR class
  description 667
VCCMD class
  description 667
VMNDR class
  description 667
VMPOSIX class
  description 668
VMRDR class
  description 668
VMSEG class
  description 668
W
WAACCNT suboperand
  ADDUSER command 87
  ALTUSER command 179
WAADDR suboperand
  ADDUSER command 87
  ALTUSER command 179
WADEPT suboperand
  ADDUSER command 88
  ALTUSER command 180
WANAME suboperand
  ADDUSER command 88
  ALTUSER command 180
warning indicator
  searching for resources with 560
warning message
  number of days before password expires 614
WARNING operand
  ADDSD command 45
  ALTSD command 105
  DELDSD command 197
  LISTDSD command 221
  PERMIT command 270
  SEARCH command 564
volume serial number (continued)
  specifying when creating access list 270
  using as search criteria 565
  using to locate model profile 40, 492
VOLUME suboperand
  TARGET command 642
VSAM data set
  protecting 652
  searching for 559
VSAM operand
  SEARCH command 559
VXMBR class
  description 668
VOLUME operand
  TARGET command 642
VSAM operand
  SEARCH command 559
VTAM (Virtual Telecommunications Access Method)
  general resource class 662
VTMAPPL class
  description 662
VXMBR class
  description 668

706 z/OS V1R11.0 Security Server RACF Command Language Reference
WHEN TIME operand
  ADDUSER command  87
  ALTUSER command  178
  RALTER command  454
  RDEFINE command  509
WHEN(APPCPORT) operand
  PERMIT command  270
WHEN(CONSOLE) operand
  PERMIT command  271
WHEN(CRITERIA) operand
  PERMIT command  271
WHEN(JESINPUT) operand
  PERMIT command  271
WHEN(PROGRAM) operand
  PERMIT command  272
SETROPTS command  623
WHEN(SERVAUTH) operand
  PERMIT command  272
WHEN(SYSID) operand
  PERMIT command  273
WHEN(TERMINAL) operand
  PERMIT command  273
WIMS class
  description  664
WITHLABEL operand
  RACDCERT ADD command  289
  RACDCERT GENCERT command  335
  RACDCERT MAP command  369
  RACDCERT REKEY command  375
WORKATTR operand
  ADDUSER command  87
  ALTUSER command  179, 181
  LISTUSER command  246
WORKATTR segment
  group profile
    displaying  246
WORKSPACE operand
  TARGET command  641
WRITER class
  description  662, 668

X
XCSFKEY class
  description  664
XFACILIT class
  description  662
XRFSOFF suboperand
  ALTUSER command  131

Z
z/OS Basic Skills information center  xvi
z/OS Network Authentication Service
general resource classes  665
z/OS UNIX
general resource classes  666
Readers’ Comments — We’d Like to Hear from You

z/OS
Security Server RACF
Command Language Reference

Publication No. SA22-7687-14

We appreciate your comments about this publication. Please comment on specific errors or omissions, accuracy, organization, subject matter, or completeness of this book. The comments you send should pertain to only the information in this manual or product and the way in which the information is presented.

For technical questions and information about products and prices, please contact your IBM branch office, your IBM business partner, or your authorized remarketer.

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 state on this form.

Comments:

Thank you for your support.
Submit your comments using one of these channels:
• Send your comments to the address on the reverse side of this form.
• Send your comments via e-mail to: mhvrdfs@us.ibm.com

If you would like a response from IBM, please fill in the following information:

Name
Address
Company or Organization
Phone No.
E-mail address
Readers’ Comments — We’d Like to Hear from You
SA22-7687-14

NO POSTAGE NECESSARY IF MAILED IN THE UNITED STATES

BUSINESS REPLY MAIL
FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

IBM Corporation
MHVRCFS, Mail Station P181
2455 South Road
Poughkeepsie, NY
12601-5400